Beyond Independent Optimization: Compression, MoE Routing, and Quantization Interactions in Multimodal Edge Intelligence

Anonymous authors
Paper under double-blind review

Abstract

Efficient multimodal inference is increasingly constrained not only by model quality or FLOP count, but by the cost of preserving, moving, routing, caching, and quantizing multimodal representations under low latency, memory, and energy budgets. This survey revieed the recent works on efficient vision-language and multimodal large language models across visual token redundancy and compression, MLLM and video token management, KV-cache optimisation, Mixture-of-Experts routing and serving, low-bit quantization, edge deployment, and hardware-aware benchmarking. The central argument is that these techniques cannot be treated as independent optimisations. Visual token compression changes the evidence and feature distributions seen by downstream language layers and MoE routers; routing instability changes expert traffic and quantization sensitivity; quantized router logits can alter expert assignment; KV-cache policies determine which multimodal evidence remains available during generation; and edge hardware constraints can turn local FLOP savings into memory movement, cache pressure, or evidence loss. We therefore organize the literature around a failure-propagation chain that makes the interfaces between compression, routing, quantization, cache management, and hardware execution explicit. Beyond cataloging methods, the survey synthesizes recurring design tensions in deployable multimodal systems: accuracy versus token budget, static compression versus content-adaptive retention, sparse routing efficiency versus expert collapse, low-bit serving versus modality-specific degradation, and benchmark convenience versus end-to-end edge realism. We further introduced a Temporal Routing Consistency as a diagnostic for video MoE models and identify open problems in routing-aware compression, cross-modal cache retention, proactive failure detection, hardware-aware co-design, and unified multimodal edge benchmarking.

1 Introduction

The rapid scaling of vision-language models has produced systems of remarkable capability, but capability has come at a cost that is no longer simply computational: it is systemic. Models that jointly process high-resolution image tiles, dense video frame sequences, OCR spans, and long text contexts routinely generate token sequences of tens of thousands of elements. Inference on consumer or edge hardware therefore requires visual token compression, KV-cache optimisation, Mixture-of-Experts (MoE) routing, quantization, and hardware-aware serving to operate together rather than as independent optimisations.

This survey's central finding is that these techniques have coupled failure modes. Visual token compression changes the distribution seen by downstream language layers and MoE routers; routing instability changes which experts receive compressed tokens; quantization perturbs both activation statistics and router logits; and edge-memory constraints force cache eviction or expert offloading that can remove information needed later in the reasoning chain (Shao et al., 2026; Liu et al., 2025a; Fu et al., 2026; Zhong et al., 2026; Yi et al., 2025b). This creates a failure propagation pathway, whereby an intervention intended for memory or latency savings at one stage becomes a distribution shift, a routing error, a quantization fault, or an evidence-deletion issue at the next. The existing literature often treats these stages independently, but effective deployment demands solving their interactions jointly.

Three converging pressures make this problem urgent. First, *token density*: one image at size 336×336 processed by CLIP-ViT-L/14 with a patch size of 14×14 creates 576 tokens, while tiling techniques produce thousands. Visual encodings contain less average information per token than text targets (Shao et al., 2026); however, deleting or merging those tokens changes the representation on which downstream routing, quantization, and caching policies were learned. Second, *sparse-model overhead*: in MoE architectures, only a fraction of parameters is active per token, but that sparsity creates irregular memory access, load imbalance, and collapse risk (Liu et al., 2025a; Sha, 2026). When compression precedes routing, this fragile balance is further perturbed by altered token statistics. Third, *deployment constraints*: mobile platforms (Li et al., 2025d), edge AI accelerators (Chitty-Venkata et al., 2024), and time-sensitive energy-constrained embedded systems impose tight constraints on latency, memory usage, and power consumption. As inference energy usage may scale super-linearly with the level of accuracy achieved, efficiency becomes a primary concern (Yang et al., 2024).

The review traces this coupled landscape from foundational redundancy principles in Vision Transformers (He et al., 2022; Jaegle et al., 2021; Tong et al., 2022) through KV-cache management, MoE routing, quantization, edge deployment, and hardware-aware evaluation benchmarks (Kwon et al., 2023; Tschand et al., 2025; John et al., 2024; Sukthanker et al., 2024). The review is organized around five primary themes, summarised in Figure 1.

These five themes are visual token compression, video and temporal redundancy, KV-cache optimisation, MoE routing with quantization and serving, and edge deployment with hardware-aware benchmarking. Figure 1 groups the literature along those lines, but the paper's main argument is that these themes are not isolated topics: each one changes the operating conditions of the next.

Contributions. This survey differs from prior surveys by treating compression, routing, quantization, cache management, and hardware constraints as a connected deployment pipeline rather than as parallel technique families. Its main contributions are:

1. **A unified failure-chain taxonomy.** We connect compression-induced routing distribution shift, routing inconsistency under low-bit quantization, and modality-heterogeneous quantization degradation into a single propagation chain. This framing explains why optimising any one stage in isolation can leave downstream failure modes unaddressed; for example, routing-aware post-training quantization recovers 1.15%–2.28% average score in low-bit MoE settings (Fu et al., 2026), while modality-aware VLM quantization retains 93.5% of FP16 performance at W3A3 (69.5 vs. 74.3) (Zhong et al., 2026).

2. **A temporal routing diagnostic for video MoE models.** We formalise *Temporal Routing Consistency* (TRC), the mean Jaccard overlap between expert sets activated by temporally adjacent frames,

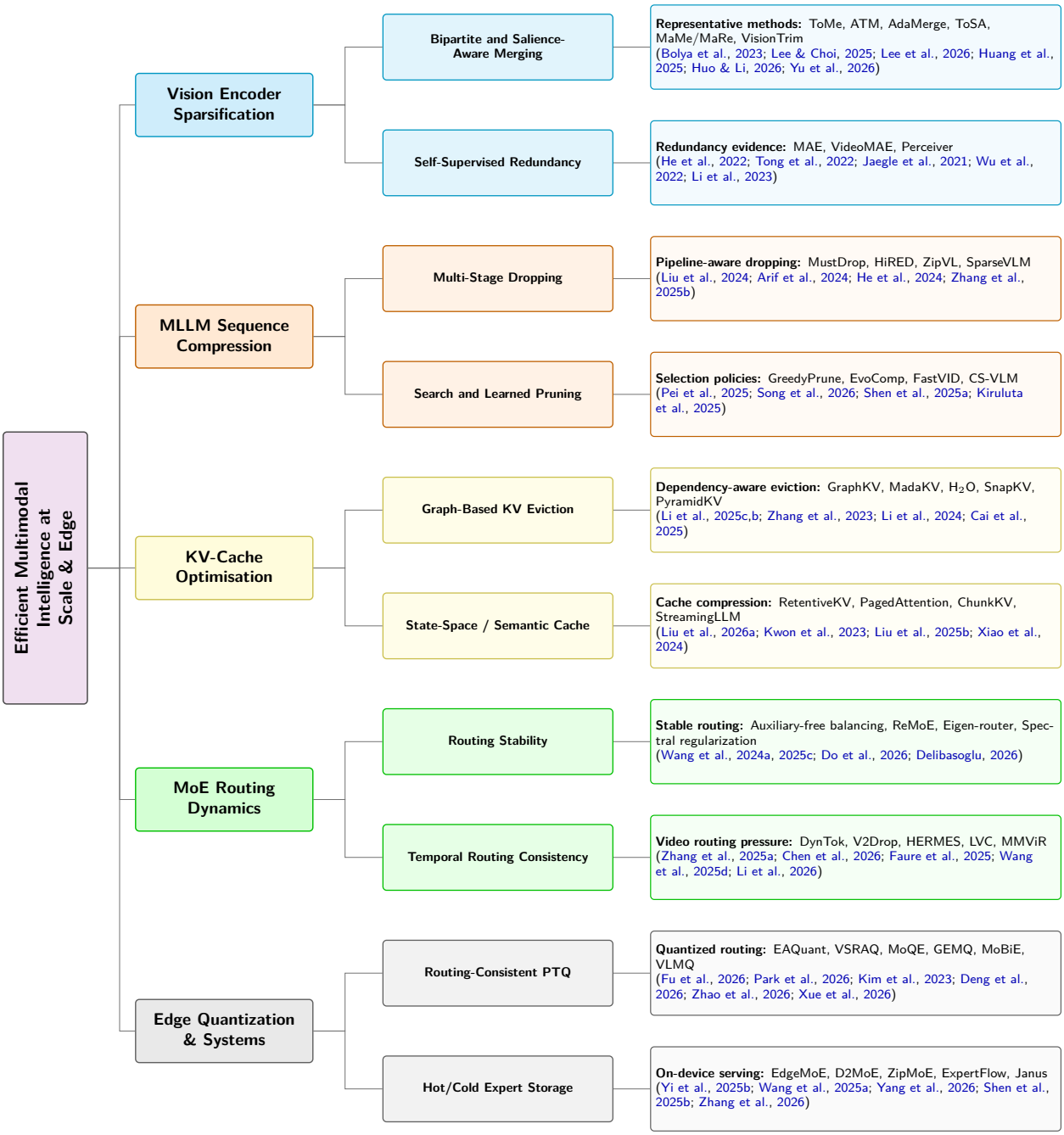

Figure 1: The decoupled taxonomy of multimodal edge efficiency research. The literature is systematically categorized across five operational bottlenecks, showing the direct structural pipeline from raw visual input token compression down to hardware-constrained co-design evaluation metrics on device.

as a diagnostic for expert jitter in video MoE pipelines. TRC makes the absence of recurrent state measurable rather than treating temporal expert continuity as an implicit assumption.

3. **A co-design view of multimodal efficiency.** We characterise compression ratio, routing behaviour, quantization precision, KV-cache residency, and hardware capacity as mutually constrained variables that shape the achievable accuracy-efficiency Pareto frontier on a given deployment target.

Table 1: Master optimization matrix for efficient multimodal inference. The table maps the dominant model class to the phase where optimization is applied, the primary system bottleneck, and the survey sections where each design axis is analysed.

Model class	**Optimization phase**	**Dominant bottleneck**	**Representative design axis**
Vision Transformers	Encoder token processing	Quadratic attention over dense visual patches	Merge, prune, or reweight redundant spatial tokens
Multimodal LLMs	Vision-to-language prefilling	Visual-token overload inside the LLM context	Stage-aware dropping, query-conditioned sparsification, positional preservation
Video MLLMs	Temporal token construction	Redundant frames plus expert jitter across adjacent frames	Dynamic frame budgets, variation-aware dropping, temporal routing consistency
Long-context MLLMs	Autoregressive decoding	KV-cache memory growth and modality-heterogeneous attention	Graph-aware eviction, modality-adaptive cache policies, state-space memory
MoE Transformers	Routing and expert execution	Load imbalance, expert collapse, and irregular memory movement	Differentiable routing, batch-level balancing, expert scheduling
Quantized VLM/MoE models	Deployment calibration	Activation outliers, routing inconsistency, and modality-specific precision sensitivity	Expert-aware calibration, structure-preserving routing, mixed precision
Edge multimodal systems	On-device serving	SRAM limits, flash/DRAM transfers, energy per byte, and thermal constraints	Hot-cold storage, prefetching, cache-affinity scheduling, hardware-aware benchmarking

4. **Systematic coverage of recent efficient multimodal inference work.** We survey over one hundred contributions from 2021–2026 across visual token compression, multi-stage MLLM token management, video redundancy, KV-cache optimisation, MoE routing and serving, quantization, edge deployment, and hardware-aware benchmarking.

Compared with surveys focused on MoE inference (Liu et al., 2025a) or multimodal token compression (Shao et al., 2026), this survey emphasises the interfaces between techniques: how compressed tokens affect routing, how routing changes quantization sensitivity, how cache eviction affects multimodal reasoning evidence, and how hardware constraints determine which combinations are deployable.

Table 1 provides the high-level map for this design space: it shows that the main efficiency bottlenecks differ by model class and deployment phase, which is why a single optimisation metric cannot describe the whole multimodal pipeline. Table 2 then narrows the focus to token reduction methods and already foreshadows the later failure-chain argument: many methods save FLOPs locally, but their interface impact determines whether those savings remain safe for downstream routing, cache retention, and quantized inference. The remainder of this survey is structured as follows. Section 2 provides foundational background. Sections 4–6 cover token compression. Section 7 addresses video and temporal redundancy. Section 8 covers KV cache. Sections 9–12 cover MoE, quantization, and edge deployment. Section 13 reviews benchmarking. Section 14 identifies open problems. Section 15 concludes.

Table 2: Comparison of visual token compression methods. Speedup figures are as reported in the respective papers and are not directly comparable across different base models and hardware. The interface-impact column indicates, per the failure-chain framework of Section 3, which interface(s) the method's output most directly affects.

Method	**Approach and stage**	**Reported speedup**	**Key limitation**	**Interface impact**
ToMe (Bolya et al., 2023)	Bipartite merging in ViT encoder	2× throughput	Fixed threshold; no spatial awareness	Interface B: merged-token routing untested
ATM (Lee & Choi, 2025)	Adaptive layer merging in ViT encoder	>30% FLOP reduction	Depth-wise only; no query conditioning	Interface B: depth-wise shift, no routing evaluation
AdaMerge (Lee et al., 2026)	Salience-weighted merging in ViT encoder	∼2×	Saliency heuristic may misfire on texture-heavy inputs	Interface A/B: saliency errors propagate to router
MaMe/MaRe (Huo & Li, 2026)	Matrix-decomposition merge in ViT and diffusion models	31% latency reduction	Designed for generative models; limited MLLM evaluation	Interface B: untested for MoE routers
ToSa (Huang et al., 2025)	Spatial-aware merging in ViT encoder	Comparable to ToMe	Spatial penalty adds overhead at high resolution	Interface A: preserves spatial layout for downstream reasoning
MustDrop (Liu et al., 2024)	Multi-stage dropping in MLLM pipeline	Large FLOP reduction	Stage schedule requires per-model tuning	Interface B: dropping alters attention context per Observation 2
HiRED (Arif et al., 2024)	Early-layer saliency pruning in MLLM prefill	Substantial throughput gain	Saliency from early layers may miss semantic importance	Interface A/B: early miscalibration compounds downstream
GreedyPrune (Pei et al., 2025)	Marginal-gain selection in MLLM encoder	High accuracy at aggressive ratios	Combinatorial search cost grows with token count	Interface B: retained-set statistics shift untested
EvoComp (Song et al., 2026)	Evolutionary learned compression in MLLM	Avoids positional gaps	Additional training procedure required	Interface B: learned retention may co-adapt with router
CSVLM (Kiruluta et al., 2025)	Compressed-sensing projection in MLLM encoder	Formal sparsity guarantees	Assumes sparse visual signal; fails on dense scenes	Interface B: projected tokens are statistical artefacts
DynTok (Zhang et al., 2025a)	Dynamic frame budgeting in video MLLM	Higher action-segment quality	Complexity estimator adds per-clip overhead	Interface B + TRC: budget shifts affect frame-wise routing
V2Drop (Chen et al., 2026)	Variation-based frame dropping in video MLLM	Large latency reduction for video	One-step window; no temporal memory	Interface B + TRC: dropped frames disrupt expert continuity

2 Background and Preliminaries

Efficient multimodal deployment rests on three architectural facts: visual inputs are converted into dense token sequences, multimodal LLMs pass those tokens through language decoders, and MoE variants replace dense feed-forward computation with routed expert banks. These facts matter because they create the main efficiency levers studied in this survey: visual tokens can be compressed, expert computation can be sparsified, and hardware placement can reduce data movement. On edge devices, however, the same levers also create failure conditions: compression shifts the token distribution, sparse routing can become imbalanced under shifted inputs, and memory hierarchies can make expert and cache movement more expensive than arithmetic. This section fixes the architectural vocabulary used by the later failure analyses.

2.1 Vision Transformers and Multimodal Architectures

Vision Transformers (ViTs) apply the multi-head self-attention mechanism to sequences of image patches, treating each patch as an independent token. The spatial redundancy of this representation is substantial: masked reconstruction pre-training produces competitive visual representations even when 75%–85% of input patches are hidden, because each visible patch carries information about its neighbours (He et al., 2022). Video data exhibits even stronger redundancy, with effective pre-training possible at masking ratios near 90% because temporal continuity makes adjacent frames highly predictable (Tong et al., 2022). An architectural alternative is to avoid full quadratic attention over the input sequence altogether by cross-attending a compact latent array to arbitrary high-dimensional inputs (Jaegle et al., 2021).

Modern MLLMs couple a visual encoder with a pre-trained LLM decoder via a connector module. This architecture introduces a critical structural inefficiency: the LLM processes visual and textual tokens identically, allocating equal per-token compute to visually redundant background patches as to semantically dense text tokens. This inefficiency is the starting point for compression, but it is also the starting point for the interaction problem. When tokens are compressed before entering the LLM, the connector's projection and the LLM's attention mechanism both receive inputs whose statistical properties differ from those seen during training. The downstream effects on MoE routing and quantization behaviour are discussed in next section.

2.2 Mixture-of-Experts: Foundations and Fragility

The key innovation of MoE is replacing a single FFN at each layer with a set of N expert sub-networks and a *router*, which selects a subset (usually Top-k, with $k = 2$) for each token. The benefits from a theoretical perspective are apparent: the number of parameters scales with N but the computation per token scales with k. (Du et al., 2024). However, the efficiency achieved through this process creates structural vulnerability which is key to this survey. Inference optimization for this regime thus emphasizes route efficiency, expertise location, batching, and memory transfer over solely focusing on reducing FLOPS. (Liu et al., 2025a).

Formally, for an input token representation x, a router computes logits $h(x) = W_r x$ and selects the active expert set through Top-k gating:

$$\mathrm{Router}(x) = \mathrm{TopK}(\mathrm{softmax}(h(x)), k).$$

The gated MoE output is then

$$y(x) = \sum_{i \in \mathrm{TopK}(x)} g_i(x) E_i(x), \qquad g_i(x) = \mathrm{softmax}(h(x))_i,$$

where E_i is the i-th expert FFN and $g_i(x) = 0$ for unselected experts. Load balancing requires the empirical selection rate of each expert to remain close to the uniform allocation,

$$\mathbb{E}_x[\mathbf{1}\{i \in \mathrm{TopK}(x)\}] \approx \frac{k}{N} \quad \forall i.$$

This setup is brittle as the router function is learned from the token distribution seen during training, not a system built to be robust against distributions shifting. Three such shifts are particularly important

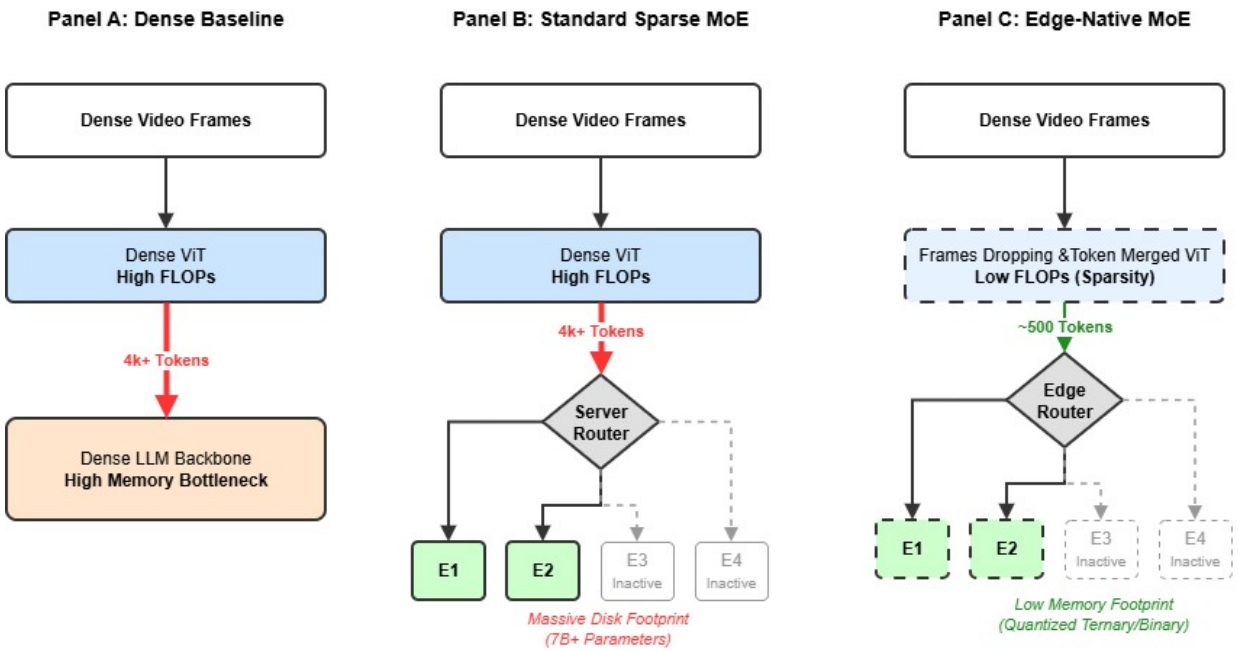

Figure 2: Architectural paradigms of multimodal efficiency. **(Panel A)** Dense baselines suffer from acute visual token inflation and high memory bottlenecks. **(Panel B)** Standard server-side MoE structures isolate active computational FLOPs but maintain prohibitive 7B+ parameter storage footprints and process uncompressed 4k+ token sequences. **(Panel C)** Edge-native MoE designs actively combine upstream token compression (merging and dropping) with downstream quantized (ternary/binary) experts to simultaneously resolve compute, VRAM, and storage constraints at the edge.

for efficient deployment of multimodal architectures. Firstly, token compression before the router causes merging of tokens and hence a different token distribution seen by the router, or dropping of some tokens from the stream completely. Secondly, quantizing the logits of the router function adds noise (Fu et al., 2026). Third, compression and quantization cause a combined shift effect that cannot be additive, since quantization distortion is added to router inputs which have themselves undergone statistical alteration via the compression process.

The primary training-time manifestation of this fragility is *load imbalance*: without regularisation, routers tend to route tokens to a small subset of experts, a failure mode termed *expert collapse* (Sha, 2026; Do et al., 2026). An expert is effectively collapsed when its utilisation rate falls below $1/N$ of the uniform allocation; for a 64-expert model, this threshold is 1.5625%. Empirically, collapse can propagate layerwise, with roughly one-third of sparse MoE layers drifting toward single-expert behaviour if routing is left poorly controlled (Sha, 2026). Classical auxiliary load-balancing losses address collapse through explicit regularisation, but they introduce training instability and loss-weight tuning; recent work therefore explores auxiliary-loss-free balancing (Wang et al., 2024a). Understanding how compression and quantization interact with these already-fragile routing dynamics is one of the central open problems this survey identifies.

2.3 Edge Deployment: A Hierarchy of Constraints

Deploying large models on resource-constrained hardware is not simply a scaled-down version of cloud serving. It is a qualitatively different problem governed by a hierarchy of binding constraints that interact with every upstream design decision.

At the tightest level, SRAM capacity determines what can reside in fast on-chip memory during a single forward pass. Mobile SoCs typically expose only a few to tens of megabytes of such storage, far below the expert FFN weights of even moderately sized MoE models, so expert parameters must be dynamically loaded

from slower memory. At the next level, DRAM bandwidth determines how quickly those weights can be transferred: mobile-class bandwidth is far below server-GPU bandwidth, making memory movement rather than arithmetic the dominant energy and latency cost (Chung et al., 2026). This has direct implications for quantization strategy, because reducing precision reduces transfer volume approximately proportionally. At the outermost level, the energy budget and thermal envelope constrain sustained throughput; peak-performance measurements can therefore overstate what a mobile or embedded device can maintain after throttling.

These constraints interact with upstream choices in non-obvious ways. Aggressive token compression reduces the number of router decisions per forward pass and can therefore reduce the number of expert transfers. But if compression disrupts routing locality, previously co-activated experts may diverge, lowering expert-cache hit rate and increasing effective transfer volume despite fewer tokens. Quantization reduces per-weight transfer cost, but it can also change which experts are selected, disrupting prefetch predictions that depend on routing stability (Fu et al., 2026). These hardware-level interaction effects make the co-design problem non-decomposable: optimising compression, routing, and quantization independently on server hardware and then deploying the combined system on edge hardware does not necessarily produce the expected efficiency. Evaluation therefore separates into mobile compressed-model benchmarking (Li et al., 2025d), architecture-level accelerator simulation (Sukthanker et al., 2024), direct power-performance characterisation (John et al., 2024), and energy-aware measurement spanning μWatt embedded targets to MWatt datacentres (Tschand et al., 2025). Figure 2 summarises the resulting architectural progression from dense baselines to server-side MoE models and edge-native sparse systems that combine compression with low-precision experts.

3 Cross-Technique Interactions and the Efficiency Co-Design Problem

This section establishes a unified framework for understanding the interactions among token compression, MoE routing, quantization, and edge deployment. The three pressures introduced in the Introduction reappear here as four operational stages: token density motivates compression, sparse-model overhead unfolds through routing and precision decisions, and deployment constraints culminate in hardware execution, with the interfaces marking where these pressures begin to interact. Each technique addresses a distinct computational bottleneck: token count, active parameter count, numerical precision, or hardware resource constraints; and has consequently been studied under separate optimization objectives and evaluation protocols. In deployed multimodal systems, however, these techniques do not operate independently. Instead, they form a sequential inference pipeline in which the output distribution of one stage becomes the input distribution of the next. As a result, optimization decisions made at one stage can induce downstream effects, including distribution shift, routing instability, quantization errors, and hardware resource contention. We refer to this causal dependency structure as the *failure propagation chain*, which serves as the organizing framework for the remainder of this survey.

3.1 The Failure Propagation Chain

Figure 3 illustrates the four-stage inference pipeline and highlights the three critical interfaces at which cross-technique failures may emerge. The KV cache is drawn outside the main left-to-right chain because it is not an independent transformation stage; instead, it is a persistent state created during inference whose footprint later couples quantized inference to hardware execution.

Formally, the inference process can be represented as the composition of four parameterized stages:

$$\mathcal{P} \;=\; \mathcal{H} \circ \mathcal{Q}_{\boldsymbol{\pi}} \circ \mathcal{R}_{\boldsymbol{\theta}} \circ \mathcal{C}_{\boldsymbol{\phi}}, \tag{1}$$

where $\mathcal{C}_{\boldsymbol{\phi}}$ is the compression operator with parameters $\boldsymbol{\phi}$ (compression ratio, merging strategy, positional encoding policy), $\mathcal{R}_{\boldsymbol{\theta}}$ is the routing operator with parameters $\boldsymbol{\theta}$ (routing temperature, load balancing objective, expert capacity), $\mathcal{Q}_{\boldsymbol{\pi}}$ is the quantization operator with parameters $\boldsymbol{\pi}$ (bit-width per component, calibration set, precision allocation policy), and $\mathcal{H}$ is the hardware execution context (memory hierarchy,

bandwidth, thermal envelope). Most existing studies implicitly assume that the optimization of $(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\pi})$ can be performed independently:

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \mathcal{L}(\mathcal{C}_{\boldsymbol{\phi}}), \quad \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{R}_{\boldsymbol{\theta}}), \quad \boldsymbol{\pi}^* = \arg\min_{\boldsymbol{\pi}} \mathcal{L}(\mathcal{Q}_{\boldsymbol{\pi}}). \tag{2}$$

The central argument of this survey is that Equation (2) holds only under an assumption of statistical independence between stages. Specifically, it requires that token compression does not alter the input distribution seen by the router and that routing decisions do not affect the sensitivity profile of subsequent quantization. These assumptions rarely hold in practical multimodal systems. Consequently, local optimizations may propagate and amplify errors throughout the inference pipeline. The remainder of this section examines these dependencies in detail. Each interface is characterised by three quantities: *what it transfers* (the information transferred between stages), *how it can fail* (the mechanisms through which failures arise), and *what has and has not been studied* (the extent to which these interactions have been studied in the existing literature.).

Interface A - Visual Input to Compression. *Transfer:* A full-resolution visual token sequence containing spatial structure, positional information, and contextual dependencies. *Failure modes:* Salient visual tokens may be incorrectly removed or merged, spatial coherence may be disrupted by similarity-based compression strategies, and positional information may be distorted, degrading downstream spatial reasoning. *State of literature:* This interface has been extensively studied in isolation, particularly in the context of token-density analysis, Vision Transformer (ViT) compression techniques, and MLLM sensitivity studies (Sections 4–6). However, the downstream impact of compression-induced representation changes on MoE routing behavior remains largely unexplored. This gap motivates the analysis of Interface B and the MoE routing analysis in Section 9.

Interface B - Token Compression to MoE Routing. *Transfer:* A compressed token sequence whose length, feature distribution, or both differ from the distribution observed during router training. *Failure modes:* Compression-induced distribution shift can degrade load balancing, increase expert-collapse tendencies, and disrupt expert specialization when merged representations occupy regions of the feature space not encountered during training. *State of literature:* This interface remains largely uncharacterized. Among all cross-technique interactions considered in this survey, it represents the most significant research gap. Section 3.2 examines the implications of compression-induced routing instability and its consequences for efficient multimodal inference.

Interface C - MoE Routing to Quantized Inference. *Transfer:* Expert activation patterns, including selected experts and their associated gating weights, which determine the weight tensors that must be loaded and the precision at which they are executed. *Failure modes:* Quantized router logits may alter expert assignments, leading to routing inconsistency and measurable performance degradation. Recent studies show that routing-aware post-training quantization can recover 1.15%–2.28% average score on low-bit MoE benchmarks, indicating that these routing perturbations are not negligible (Fu et al., 2026). Additional failure modes include expert-importance miscalibration and interactions between routing-induced activation patterns and modality-specific quantization outliers. *State of literature:* This interface has been partially investigated in isolation (Fu et al., 2026; Park et al., 2026). However, the extent to which routing instability originating at Interface B amplifies quantization errors at Interface C remains unknown. Section 3.3 explores this interaction.

3.2 Interface B: The Compression-Routing Interaction

Interface B is the least studied of the three and structurally the most consequential, because it is the earliest point at which cross-technique failure originates and failures here propagate through every downstream stage.

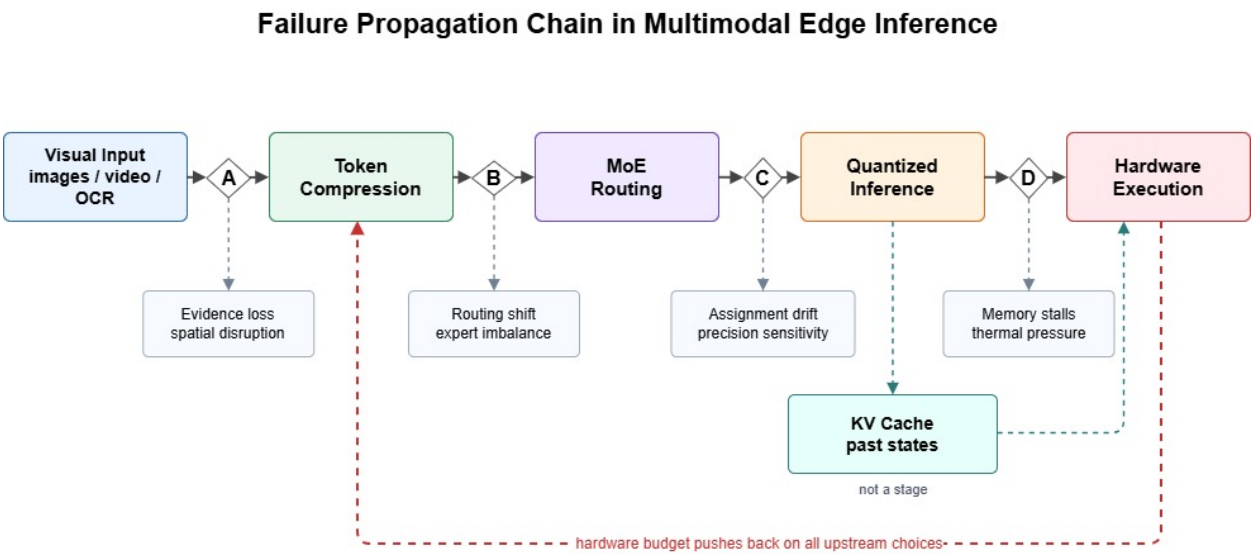

Figure 3: Failure propagation chain for efficient multimodal edge inference. Each stage receives the output distribution produced by the previous stage, so local efficiency optimisations can become downstream failure modes: compression may remove visual evidence, compressed representations may shift MoE routing, quantized router logits may change expert assignments, and hardware limits may feed back into compression, routing, and precision choices. The KV cache is shown outside the main chain because it stores past states created during inference rather than acting as a separate processing stage; nevertheless, its size and retention policy convert upstream token, routing, and precision decisions into later memory stalls and hardware pressure.

3.2.1 How Compression Shifts the Router's Input Distribution

Token compression operates on the visual token sequence before or during the LLM prefill stage. The two dominant paradigms affect the router's input distribution through distinct mechanisms:

Token dropping removes tokens entirely, reducing sequence length. The surviving tokens retain their original representations, so their *marginal* distribution is unchanged relative to training. However, the *joint* distribution of the sequence changes: dropped tokens contributed to the multi-head self-attention context that shapes the key and value projections of surviving tokens in later encoder layers. When those tokens are absent, the surviving tokens' contextualised representations differ from those produced at training time. The magnitude of this context shift grows with the compression ratio and is largest when dropped tokens are spatially adjacent to retained ones precisely the regime targeted by background-dropping methods.

Token merging replaces pairs of similar tokens with their weighted average. Merged tokens are statistical artefacts: they are convex combinations of source tokens whose representations may occupy low-density regions of the router's learned token space. The router, trained on a distribution of individual patch embeddings, has no explicit mechanism for handling such combinations. In the best case, merged tokens are routed similarly to their source tokens and the routing distribution is approximately preserved. In the worst case when merged tokens fall near decision boundaries between expert subspaces, small perturbations in the merge weights produce large changes in routing assignments, amplifying sensitivity to the merging hyperparameters.

3.2.2 Bounding the Magnitude of the Interaction

The magnitude of compression-induced routing disruption has not been directly measured in the literature. The closest evidence comes from recent studies on routing-consistent quantization.

Fu et al. (2026) demonstrate that quantization-induced perturbations to router logits are sufficient to disrupt top-k expert assignment, and that preserving routing consistency via KL divergence alignment between full-precision and quantized routing probabilities recovers accuracy on reasoning benchmarks across three MoE architectures. Park et al. (2026) independently show that explicitly constraining quantized routing decisions to match full-precision routing improves accuracy beyond standard calibration procedures. Together, these results establish that even the relatively small perturbations introduced by post-training quantization are sufficient to produce measurable, task-relevant routing changes.

Compression introduces perturbations through two channels that differ from quantization. First, token merging modifies token representations directly, potentially moving them away from the training distribution. Second, token dropping changes the attention context available to surviving tokens, altering their contextualized representations. Unlike quantization, which primarily perturbs activation magnitudes, compression can modify both representations and sequence structure. These observations suggest that compression may induce routing shifts comparable to, or potentially larger than, those observed under quantization. However, this remains a hypothesis rather than an established result. Quantifying routing assignment changes under compression is therefore an important open problem and a necessary step toward compression-aware MoE design. This is identified as a priority open problem in Section 14.

3.2.3 Implications for Routing Objectives

Current token-compression methods are typically evaluated using accuracy-efficiency trade-offs such as FLOPs reduction, latency, and task performance. To the best of our knowledge, routing stability is not reported as a standard evaluation metric. This omission is notable because compression directly alters the inputs to the router. A routing-aware compression objective should therefore incorporate a stability term that penalizes deviations between routing distributions on compressed and uncompressed inputs. One natural formulation is to minimise the KL divergence between the corresponding expert-assignment distributions, analogous to routing-consistency objectives used in recent quantization studies.

3.3 Interface C: Routing Instability and Quantization Sensitivity

Interface C captures the interaction between routing decisions and quantization. The relationship is bidirectional: quantization can alter routing by perturbing router logits, while routing behaviour influences quantization by determining which experts require higher precision. The first direction is partially characterised (Fu et al., 2026; Park et al., 2026); whereas the latter remains largely unexplored.

3.3.1 How Routing Decisions Shape Quantization Sensitivity

Several recent quantization methods allocate precision according to expert importance, assigning higher bit-widths to experts that receive more routing traffic during calibration (Deng et al., 2026). This strategy assumes that expert activation patterns remain stable between calibration and deployment. However, this assumption may not hold when routing distributions shift for two resons.

First, if the set of high-traffic experts at calibration time differs from the set at deployment time due to compression-induced distribution shift at Interface B the precision allocation is miscalibrated to the actual deployment routing. The experts that receive degraded precision are not the least important ones in absolute terms; they are the ones whose importance was underestimated at calibration because the compressed input distribution suppressed their activation frequency.

Second, the experts most affected by this miscalibration are low-frequency experts—those activated only by rare or highly specific input patterns. These are precisely the experts that routing inconsistency research identifies as most informationally critical. Fu et al. (2026) show that the assignments that change under W4A8 quantization disproportionately involve low-frequency expert-token combinations with high information content. A compression policy that further suppresses the activation of these experts during calibration therefore creates a compounding failure: the most important rare-expert activations receive both unstable routing and insufficient quantization precision.

3.3.2 The Amplification Relationship

We characterise the joint failure as an *amplification relationship*: routing instability introduced at Interface B increases the probability that Interface C failures already present under quantization alone will manifest and produce observable accuracy degradation.

Let Δ_{Q} denote the routing assignment change rate under quantization alone, and let Δ_{C} denote the additional assignment change rate induced by upstream compression. The joint change rate Δ_{CQ} satisfies: $\Delta_{\mathrm{CQ}} \geq \Delta_{\mathrm{Q}} + \Delta_{\mathrm{C}} - \Delta_{\mathrm{Q}} \cdot \Delta_{\mathrm{C}}$, where the right-hand side is the inclusion-exclusion lower bound under the assumption that the two perturbation sources affect disjoint subsets of assignments. In practice, both perturbation sources are more likely to affect the same marginal assignments those closest to decision boundaries so the actual joint change rate likely exceeds this bound. We discuss a first-order characterisation and a tighter bound would require empirical measurement of the correlation between compression-induced and quantization-induced routing changes, which does not yet exist in the literature and is identified as an open problem in Section 14.

3.4 The Non-Decomposable Pareto Frontier

The interaction effects at Interfaces B and C have a direct consequence for the efficiency optimisation problem. Compression, routing, and quantization are typically studied as separate accuracy-efficiency trade-offs. Compression reduces token count at the cost of information loss, routing balances expert utilization against computational overhead, and quantization reduces memory and energy consumption at the cost of numerical precision.

Most existing optimization strategies implicitly assume that these trade-offs can be optimized independently and then combined. However, the interactions discussed in previous sub-sections suggest that this assumption is often invalid. Compression changes routing behaviour, routing influences quantization sensitivity, and quantization can in turn affect routing decisions. As a result, the optimal operating point of the full system cannot generally be obtained by independently optimizing its components. A configuration that appears optimal when evaluated in isolation may become suboptimal once deployed within the complete inference pipeline. The discrepancy arises from cross-stage dependencies rather than poor parameter tuning.

This observation motivates a shift from component-level optimization toward joint or co-designed optimization strategies that explicitly account for interactions among compression, routing, and quantization. The discrepancy is not a tuning failure; it is a structural consequence of ignoring cross-stage dependencies. The only remedies are joint optimisation of the pipeline parameters or, at minimum, sequential optimisation that holds upstream parameters fixed while tuning downstream ones.

3.4.1 Hardware Constraints as Additional Pareto Pressure

On edge hardware, the non-decomposability of the Pareto frontier is further tightened by hardware-specific constraints that interact with all three upstream stages. The energy cost of a forward pass on edge hardware is dominated by memory transfer volume, not arithmetic throughput (Chung et al., 2026). This transfer volume depends jointly on: quantization precision (lower precision reduces bytes per weight), expert activation patterns (which experts are loaded determines transfer volume), and token count (which determines how many routing decisions and expert activations occur per forward pass). These quantities are jointly determined by the pipeline parameters $(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\pi})$ and cannot be minimised independently.

A compression policy that reduces token count but disrupts routing locality causing previously co-activated experts to diverge may *increase* net transfer volume despite fewer tokens, because it forces loading of a more diverse set of expert weights per forward pass. Similarly, a quantization policy that reduces per-weight transfer cost but destabilises routing may increase transfer volume by forcing prefetch misses on experts that routing predictions failed to anticipate. These are concrete, measurable hardware-level consequences of the interaction effects at Interfaces B and C, and they are invisible to any evaluation methodology that benchmarks compression, routing, and quantization on separate hardware runs.

3.5 Temporal Routing Consistency: A Diagnostic for Video MoE Models

The interaction effects described above apply to all multimodal MoE models, but they take a specific and qualitatively distinct form in video understanding models. Video MoE models process temporally ordered token streams, and the failure chain acquires a temporal dimension that static image models do not exhibit. This subsection introduces a diagnostic metric for this temporal dimension.

3.5.1 The Temporal Continuity Problem in MoE Models

Recurrent architectures such as LSTMs guarantee representational continuity across time steps by construction: the hidden state h_t is computed as a function of h_{t-1}, ensuring that the representation of frame t is explicitly conditioned on the representation of frame $t-1$. Transformer-based MoE models provide no analogous guarantee. The router assigns experts to tokens independently: given a token representation $\mathbf{x}_t$, the router selects experts based solely on $\mathbf{x}_t$'s current features, without knowledge of which experts were activated for temporally adjacent tokens.

This independence is a feature for static inputs it enables parallelism and hardware efficiency but it is a potential failure mode for temporally ordered sequences. If consecutive frames are routed to entirely different expert subsets, the expert-specific representations for frames t and $t+1$ reside in different activation subspaces, disrupting the representational continuity that temporal reasoning requires. The shared token embedding space provides some continuity, but this is a soft constraint: routing collapse (Sha, 2026) or routing instability introduced by upstream token compression can disrupt even this implicit coherence.

3.5.2 Temporal Routing Consistency as a Metric

Let $E_t \subseteq \{1, \ldots, N\}$ denote the set of experts activated by all visual tokens from frame t under Top-k routing, where N is the total number of experts and T is the number of frames in the clip. The *Temporal Routing Consistency* (TRC) score is the mean Jaccard overlap between expert activation sets for all pairs of temporally adjacent frames:

$$\mathrm{TRC} = \frac{1}{T-1} \sum_{t=1}^{T-1} \frac{|E_t \cap E_{t+1}|}{|E_t \cup E_{t+1}|}. \tag{3}$$

$\mathrm{TRC} \in [0, 1]$, with $\mathrm{TRC} = 1$ indicating perfect expert overlap between all adjacent frames (maximum temporal continuity) and $\mathrm{TRC} = 0$ indicating completely disjoint expert activation across all adjacent frames (maximum expert jitter).

TRC under compression. The interaction between Interface B and TRC is direct: upstream token compression that disrupts routing locality will lower TRC by causing adjacent frames' compressed representations to be routed to more divergent expert subsets. A compression method that degrades TRC should be considered harmful for video understanding tasks even if its static accuracy-FLOPs tradeoff appears favourable, because the TRC degradation indicates a loss of temporal coherence that single-frame accuracy metrics do not capture. Jointly optimising compression and TRC would provide a principled training objective for video multimodal MoE models that is temporally coherent and routing-stable under compression. To the best of our knowledge, no existing method pursues this joint objective.

3.6 Implications of the Failure Chain

The failure propagation chain and the three interfaces characterised above provide the organising framework for the remainder of this survey. The KV cache sits outside this four-stage diagram for the same reason it sits outside Figure 3: it is not a fifth processing stage. Instead, it stores intermediate states produced by the chain, and those stored states determine how earlier decisions reappear later as decoding-time memory pressure. Each subsequent section covers one or more stages of the chain; Table 3 maps survey sections to chain stages and identifies the specific interface interaction each section informs.

Three implications of the chain analysis are as follows:

Table 3: Mapping of survey sections to failure chain stages. The Interaction informed column identifies which cross-stage interface each section's content characterises or motivates.

Section	Chain stage	Dominant bottleneck	Interaction informed
§4	Pre-compression evidence	Uneven information density	Interface A
§5	Stage 1: Compression	ViT spatial redundancy	Interface A, Interface B
§6	Stage 1: Compression	MLLM pipeline sensitivity	Interface A, Interface B
§7	Stage 1 + TRC	Temporal redundancy, expert jitter	Interface B + TRC
§8	Cache-to-reasoning boundary	KV memory growth, evidence retention	Cache pressure + hardware feedback
§9	Stage 2: Routing	Load imbalance, collapse	Interface B, Interface C
§10	Routing-to-systems boundary	Expert movement, cache interference	Interface C + hardware feedback
§11	Stage 3: Quantization	Activation outliers, routing drift	Interface C
§12	Stage 4: Hardware	SRAM, bandwidth, energy	Model-to-hardware Pareto constraint
§13	Evaluation	Benchmark gaps	All interfaces

Implication 1: Compression quality has a routing dimension. Current compression benchmarks measure accuracy and FLOPs. They do not measure routing stability under compression. Section 5 and Section 6 review compression methods through this additional lens, noting where routing implications can be inferred from existing results and where they remain uncharacterised.

Implication 2: Routing evaluation should condition on the compression policy. Current routing benchmarks measure load balance and expert utilisation on uncompressed inputs. A routing method that achieves good balance on full-resolution tokens may fail under compressed inputs with shifted statistics. Section 9 reviews routing methods with this conditionality in mind.

Implication 3: Quantization calibration should account for upstream compression. Expert-importance quantization is calibrated on the token distribution seen during calibration runs. If those calibration runs use uncompressed inputs but deployment uses compressed inputs, the calibration is systematically mismatched. Section 11 identifies where existing quantization methods are and are not robust to this mismatch.

4 Token Redundancy in Multimodal Transformers

Token redundancy is the starting point for almost every efficiency method in multimodal transformers. The topic asks which visual, temporal, or cross-modal tokens carry information that is predictable from context and can therefore be removed, merged, or down-weighted. Its importance for edge deployment is direct: redundant visual tokens expand every downstream cost, including encoder attention, LLM prefilling, KV-cache memory, and MoE routing traffic. The failing mode here is that redundancy should not be viewed as irrelevance, as even low density patches, OCR spans, or frames can become important to support grounding or reasoning. This chapter thus approaches redundancy as something that should be quantified on a case-by-case basis, not as an opportunity to indiscriminately remove tokens.

4.1 Self-Supervised Evidence

Quantitative proof of the visual token redundancy can be seen most clearly in reconstruction-based pre-training. If up to 75%-85% of patches can be masked and yet competitive features still learned, this means that much of the visible sequence is predictable based on context. (He et al., 2022). In this regard, video further reinforces the claim by showing that masking ratios around 90% work effectively as a consequence

of temporal continuity, which is another form of redundancy (Tong et al., 2022). Collectively, these findings demonstrate a quantitative baseline for the claim: *at least* 75% *of visual tokens can be removed without any information loss*. Instead of detecting redundancy as a post-hoc process, latent array-based models eliminate full-resolution token representations from the beginning while using iterative cross-attention to condense the input. (Jaegle et al., 2021).

4.2 Cross-Modal Redundancy

In the case of video, an extra kind of redundancy arises, since most video tokens do not semantically align with any text token — they encode background, irrelevant transitions, and redundant information. Redundancy awareness for coarse pre-training mitigates alignment effort for non-relevant visual tokens and enhances retrieval by reducing noisy signals generated by non-aligning video tokens (Wu et al., 2022). Answering questions demands fine alignment, since many frames do not contain information relevant to the specific question posed even if those frames have saliency independent of the task. Question-guided frame selection helps focus computation on relevant frames that influence the answer (Li et al., 2023). This progression from sentence-to-video alignment to query-to-token alignment corresponds to the move from global token constraints in ViT to query conditioning in MLLMs.

4.3 The Information Density Framework

Recently, new theories have brought together the three forms of redundancy mentioned above under the heading of *token information density*, which is defined as the information content of a token when conditioned on its context (Shao et al., 2026). When the token's information density drops below a certain threshold, it may be eligible for elimination; however, both the threshold and the density itself depend on the modality (visuospatial tokens have lower average densities compared to text), and the context (background tokens tend to have lower density than object boundaries). The decision to apply these theories has become a choice between eliminating low density tokens or keeping them but with diminished weight; attention-based learning is a way to implement deletion, while position rescaling achieves the opposite conservatively (Mahajan et al., 2025).

5 Token Compression in Vision Transformers

Token compression in Vision Transformers converts redundancy into raw efficiency gains within the visual encoder. The rationale behind this is that ViT self-attention depends on the number of image patches, and compressing visual sequences before they enter the LLM might lead to more efficient encoders as well as shorter multimodal context lengths. Yet compression on edge devices usually needs to be extreme, leading to unique sources of error, such as information loss due to dropping tokens, conflicts due to merging tokens, and overly harsh compression due to fixed compression limits. This is why approaches for token compression discussed in this section revolve around one key question: How does one compress tokens while not destroying the feature representation that other layers rely on?

5.1 From Token Dropping to Token Merging: A Design Dichotomy

The basic conflict in ViT token compression can be stated as irreversibility and accuracy vs. efficiency and simplicity. Token pruning, where low importance tokens are dropped completely, offers maximum computational efficiency since less input elements are processed by subsequent layers, but sacrifices information which might turn out important later. By contrast, token merging preserves overall information at the expense of generating semantically inconsistent tokens; however, no information voids are created. Training-free bipartite soft matching provides a practical balance, where tokens are split into two subsets and similar token pairs from each subset are merged, leading to a speedup of up to 2× while suffering an accuracy loss of only 0.2%–0.3% on ImageNet for DeiT, MAE and CLIP-like models (Bolya et al., 2023). The superiority of merging tokens over dropping them lies in the fact that the former technique preserves an average representation of the redundant patches while the latter causes loss of information. This is demonstrated in Figure 4.

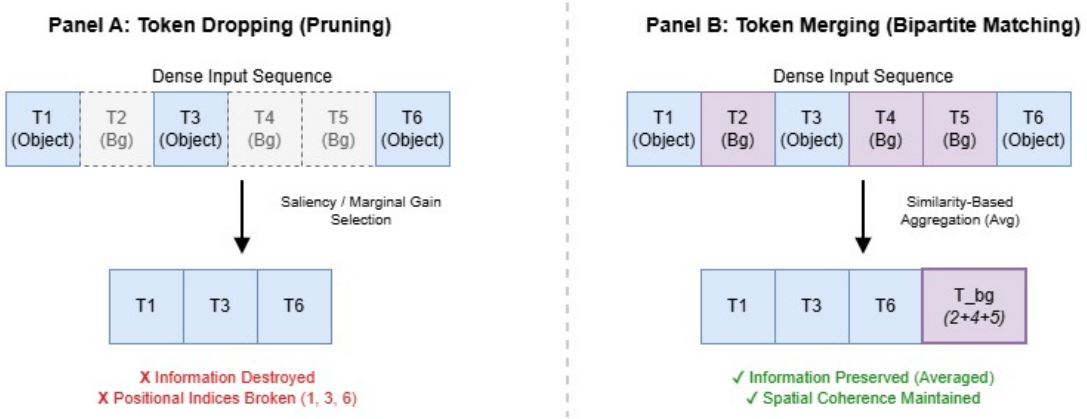

Figure 4: Token reduction mechanisms through visual tokens. **(Figure A)** The concept of token dropping refers to the permanent removal of less salient regions (e.g., background) from the input image, which increases the extent of Flops saving, but at the same time, destroys the aggregate information as well as continuity of position indexing. **(Figure B)** The token merging process makes use of bipartite matching.

5.2 Layer-Wise and Spatial Failure Modes in Token Merging

A fixed global similarity threshold creates two orthogonal failure modes. First, thresholds that are safe in late semantic layers can over-merge early layers where fine-grained texture differences still carry local information. Layer-specific adaptive thresholds address this depth-wise mismatch and can reduce DeiT FLOPs by over 30% while preserving accuracy exactly, not approximately (Lee & Choi, 2025). Second, feature similarity alone can merge visually distant regions that happen to share embedding statistics, such as sky and sea patches with similar colour distributions. Adding spatial distance to the merge criterion protects layout information and is especially important for tasks such as scene understanding and visual question answering (Huang et al., 2025). A complete merging policy must therefore satisfy both depth-wise calibration and spatial coherence; optimising only one leaves a distinct failure channel open.

5.3 Extending Compression Beyond Accuracy: Generativity and Throughput

Once basic merging is effective, the next trade-off is whether to improve the accuracy-throughput frontier by protecting salient tokens, changing the algebra of merging, or sparsifying attention without changing the token count. Salience-weighted similarity scoring protects foreground and edge tokens while adaptive per-layer ratios control how aggressively each depth is compressed, reducing high-compression Top-1 degradation to about half that of fixed matching baselines (Lee et al., 2026). A matrix-decomposition formulation shifts the bottleneck from pairwise matching to GPU-friendly batched operations and can pair token merging with restoration, reducing Stable Diffusion generation latency by 31% while preserving visual quality (Huo & Li, 2026). Attention sparsification follows a complementary axis: rather than reducing token count, it reshapes attention-score statistics and retains only significant token-token connections, lowering attention density over the existing sequence (Chen et al., 2025).

6 Token Management in Multimodal LLMs

Token management for multimodal LLMs is concerned with choosing which visual tokens to preserve, in what order, and after how many have been fed into the language model pipeline. This subject is important since the LLM considers visual and textual tokens as one single context, meaning that all saved visual tokens will take up attention compute, KV-cache, and decoding memory. In the edge failure scenario, the concern becomes stage sensitivity; compress too early and you lose visual detail, compress too late and there is very little gain in compute saved, and reindex poorly and positional information may be lost despite retaining proper visual content.

6.1 The Pipeline Sensitivity Problem: Where to Compress?

A critical question that Visual Transformers' compression does not consider is *at what point in the MLLM processing pipeline* would be the best time to apply token pruning. This is because at various stages of the process, having the same compression level comes at very different costs: encoder dropping reduces visual FLOPs while keeping computations on the LLM side intact, prefill dropping reduces memory consumption but disrupts attention patterns, and decoding dropping can be highly risky since every dropped token alters the generation probability. Multistage strategies take advantage of this asymmetry by employing aggressive dropping at encoding, moderate dropping at prefill, and mild dropping at decoding stages. (Liu et al., 2024). The other force that becomes relevant is high-resolution tiling, since a single scene may contain up to 2,000-5,000 visual tokens. Given such parameters, early attention allows us to produce an efficient saliency signal that would allow one to prune some tokens prior to executing the computationally heavy ViT layers. (Arif et al., 2024).

6.2 The Compression Approach Spectrum: Greedy, Structural, and Learned

Given a fixed number of tokens to be preserved, there are three general approaches to deciding which tokens to keep: greedily selecting the tokens based on the information gain they provide, signal-theoretic projection in which no information is lost, and learning from performance. Greedy combinatorial selection selects tokens so as to maximize marginal gain and can preserve accuracy even when the pruning ratio is very large. (Pei et al., 2025); query-conditioned sparsification creates a budget that is dependent on the ongoing query of the text by employing cross-modal attention for determining relevant visual tokens (Zhang et al., 2025b); while compressive sensing projections offer formal guarantees for the preservation of signals of the sparse visual representation, beyond empirical accuracy gains (Kiruluta et al., 2025). Finally, learned compression is another dimension that occupies the spectrum: evolutionary pseudo-labeling enables training an efficient compressor whose decisions are guided by performance of the task at hand without positional gaps created by token dropping. (Song et al., 2026). The resulting trade-off is clear: greedy, query-conditioned, and structural methods are largely training-free but less adaptive, while learned compressors adapt to task statistics at the cost of an additional training procedure.

6.3 Positional Integrity Under Compression

In all types of compression mechanisms, there exists an additional form of failure where: once the tokens have been removed or combined, the remaining tokens possess erroneous or inconsistent positional indices leading to poor performance in spatial tasks. This can be addressed by encoding spatiotemporal information in the compressed tokens rather than renumbering surviving tokens for better performance in token dropping regimes (Huang et al., 2026). In general, a training free approach that combines positionality with the use of viewpoint tiles and merging tokens based on text input becomes especially relevant during intensive compression and spatial queries. (Yu et al., 2026). A complementary alternative leaves indices intact but changes positional encoding magnitudes, amplifying informative tokens so they exert greater influence on attention scores (Huang & Yuan, 2026). Positional encoding integrity is therefore a first-class constraint in MLLM compression, not an afterthought.

7 Video Understanding and Temporal Redundancy

Video understanding extends token redundancy from space into time. The topic is important because video inputs can multiply image-token costs by hundreds or thousands of frames, making uncompressed processing infeasible for edge devices. Yet temporal data also changes the failure landscape: static background tokens are highly compressible, but motion, event boundaries, and temporally separated evidence may be lost if compression is applied frame by frame without respecting sequence structure. When MoE routing is added, another edge condition appears: adjacent frames may be sent to inconsistent expert subsets, creating expert jitter instead of temporal continuity. This section examines video compression as a temporal-reasoning problem rather than as static image compression repeated over frames.

7.1 Why Video Compression Requires More Than Static Token Reduction

Applying standard image token compression independently to each video frame misses the central structure of video data: temporal continuity. A background region that is uninformative in frame t is equally uninformative in frames $t-1$ and $t+1$, and removing it independently in each frame is wasteful—it should be identified once and suppressed throughout the clip. Conversely, a region undergoing motion is maximally informative precisely *because* it changes across frames, and suppressing it would lose the most task-relevant signal. This interplay between static redundancy and temporal variation creates compression challenges qualitatively different from the image case.

7.2 The Temporal Sequencing Challenge: MoE Parallel Routing vs. Recurrent Constraints

One significant structural challenge in using MoE-based architectures for video token streams stems from MoE routers assigning expert functions *independently* to individual tokens. Specifically, given a token embedding, the router determines which expert functions to employ according to the features of this particular token, without regard for whether any experts are activated for adjacent tokens. While this property is beneficial for processing of static data, it introduces a possible point of failure for processing of time-ordered data: if two consecutive frames are mapped to non-overlapping sets of expert functions, the respective expert-based representations will live in distinct activation spaces, disrupting representational consistency across frames necessary for temporal reasoning. This can be understood as the fundamental difference between MoE-based and recurrent-based architectures like LSTM networks (Hochreiter & Schmidhuber, 1997), where the presence of an explicit *hidden state* guarantees that the representation of frame $t+1$ is a function of the hidden state of frame t.

This can be assessed directly via the *Temporal Routing Consistency* (TRC) measure described in Section 3.5, which evaluates the mean overlap (using the Jaccard metric) between the sets of experts invoked by temporally neighboring frames. TRC is high when adjacent frames are handled using an overlapping set of experts, maintaining a consistent subspace of experts for temporal inference; otherwise, if the value is low, there is "expert jitter," i.e., adjacent frames use disjoint sets of experts, requiring temporal continuity to be recovered from the latent embedding space alone.

Practically, however, the amount of routing continuity of video tokens has yet to be characterised systematically. The fact that motion information can be extracted and disassociated from per-frame content embeddings indirectly suggests that rapid changes in motion tokens are distinct from static content tokens (Zhang & Fu, 2025). In this scenario, a well-trained MoE router will route motion tokens to specific "motion experts" and static tokens to other experts designated as "content experts," thereby roughly preserving the temporal consistency. However, such specialisation cannot always be taken for granted, since routing collapse (Sha, 2026) and routing instability resulting from token compression prior to the router (Fu et al., 2026) can impede this specialisation process. Adaptive video token compression provides some remedy against a lack of recurrent structure by assigning higher budgets to complex action segments relative to the simpler, static shots (Zhang et al., 2025a). The opposite lever is token removal on the basis of low temporal variation: feature comparison across consecutive frames allows tokens with very little temporal variation to be removed, resulting in far greater latency savings when dealing with videos as inputs compared to images due to the high amount of zero variation in the video stream (Chen et al., 2026). This approach does not benefit from the existence of recurrence, which means that it cannot rely on hidden state accumulation like an LSTM would; it is cheaper, but lacks the ability to maintain temporal context beyond a single step.

The upshot of all this is that video MoE models at present operate under the implicit assumption that there exists enough temporal coherence both in the token embedding space and in expert specialisation, unlike LSTM-based architectures, which provided such guarantees. Architectural choices can allow the testing of such an assumption via optimisation of TRC or addition of the routing consistency loss, forcing a low Jaccard overlap penalty on consecutive frames' expert selection to keep the expert jitter in check during training(see Section 14).

State-space models such as Mamba occupy a middle ground between the two extremes described above: they maintain a compressed recurrent state that is updated across the sequence, providing some of the

representational continuity that LSTMs guarantee by construction, but without the sequential step-by-step computation that limits LSTM throughput. Whether SSM-based video backbones exhibit higher TRC than attention-based MoE models—that is, whether their compressed recurrent state induces more consistent expert routing across adjacent frames than the shared-embedding-space mechanism available to Transformer-based MoE routers—is an open empirical question that the TRC diagnostic is well suited to answer.

7.3 Long-Form Video: From Frame-Level Compression to Scene-Level Abstraction

As video length extends from seconds to hours, frame-level token compression becomes insufficient: even compressing each frame to a single token would leave thousands of tokens for an hour-long video. Long-video efficiency therefore shifts from local token pruning toward hierarchical and memory-augmented abstraction (Zou et al., 2024). Scene-level hierarchies segment videos into semantic episodes, summarise each episode into a compact representation, and perform long-range temporal reasoning over the episode sequence rather than raw frames (Faure et al., 2025). This hierarchy introduces a new sequencing constraint: episode order must be preserved for causal reasoning, and summaries must retain enough scene context for downstream queries. Multi-granularity representations extend the hierarchy by combining frame-, clip-, and scene-level features into a unified multi-scale token sequence, trading richer context for higher token cost (Li et al., 2026).

The efficiency-understanding tension at the long-form scale is sharpest for real-time video. Online pruning can bound latency by using a lightweight preliminary attention pass to estimate token importance before committing to full-attention computation (Jin et al., 2025). A complementary approach compresses the temporal token sequence directly inside the VLM context window without retraining, exploiting the fact that abstract predictive video representations already encode temporal redundancy structurally and are therefore cheaper to compress (Wang et al., 2025d; Drozdov et al., 2024).

7.4 Self-Supervised Foundations for Efficient Video Representations

The efficiency of downstream compression depends critically on the structure of pre-trained video representations. Reconstruction-based pre-training with 90% masking forces each visible token to carry high information content, structurally reducing temporal redundancy before any downstream pruning is applied (Tong et al., 2022). Predicting abstract representations rather than pixels pushes this effect toward semantically organised temporal encodings that are inherently more compressible (Drozdov et al., 2024). Cross-modal redundancy can also be reduced during video-language pre-training by weakening alignment to visual regions that carry little text-relevant information (Wu et al., 2022).

8 KV Cache Optimisation

The KV cache enables efficient autoregressive generation by storing the key and value states produced by causal self-attention (Vaswani et al., 2017). This avoids recomputing attention projections for the full prefix at every decoding step, but it moves the bottleneck from arithmetic to memory residency. In multimodal edge inference this topic is crucial because image patches, video frames, OCR spans, and text tokens all leave cached states behind during generation. For a decoder with batch size B, retained token length T, number of layers N, KV heads H, head dimension d, and element size p bytes, the cache footprint is approximately

$$2BTNHdp \text{ bytes},$$

where the factor of two comes from storing both keys and values. In words, KV memory grows linearly with tokens, layers, heads, hidden dimension, precision, and batch size. Long-sequence transformer variants reduce attention complexity architecturally (Beltagy et al., 2020; Zaheer et al., 2020), and efficient-NLP surveys place such methods within a broader compression toolkit (Treviso et al., 2023); however, they do not by themselves solve the deployment-time failure condition that every retained prefix token still occupies cache memory. KV-cache optimisation therefore asks which cached states must remain exactly addressable during prefill and decoding, and which can be virtualised, compressed, or evicted before the edge memory budget is exceeded. In the terminology of Section 3, this is why KV-cache optimisation sits at the cache-to-reasoning boundary rather than inside the four-stage chain: it stores states created after upstream compression, routing,

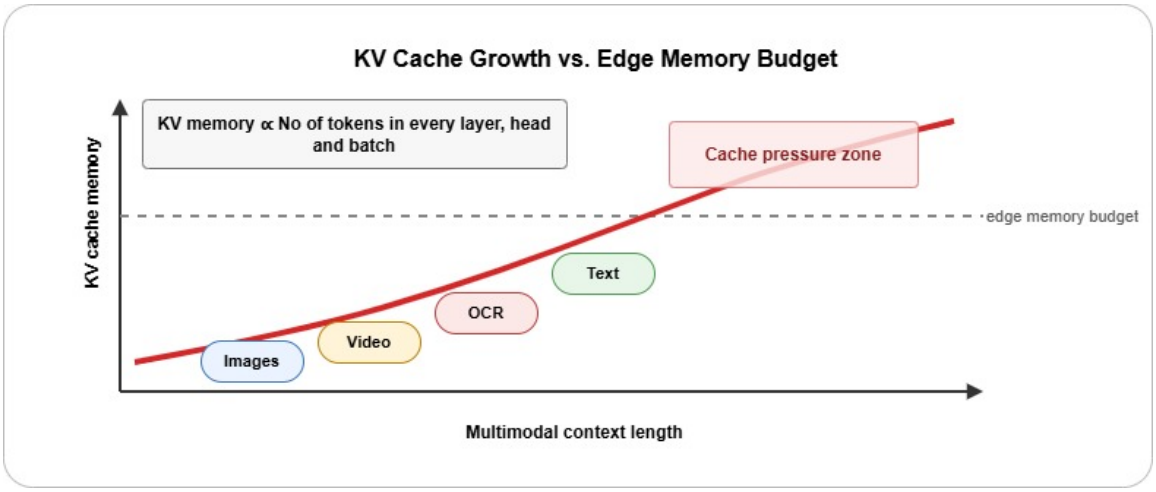

Figure 5: KV-cache growth under edge memory constraints. During autoregressive multimodal inference, each retained token contributes key and value states across transformer layers, attention heads, batch elements, and numerical precision. Therefore, the KV cache grows with the effective multimodal context length, including image patches, video frames, OCR tokens, and text tokens. The dashed horizontal line represents the available edge-device memory budget. Once the cache footprint crosses this budget, inference enters a cache pressure zone where latency, memory movement, and energy cost increase sharply. This motivates KV-cache optimisation methods that compress, evict, or selectively retain cache entries for long-context edge deployment.

and precision decisions, then turns those earlier choices into later memory and evidence-retention constraints. Figure 5 visualises this memory-pressure curve for multimodal contexts under an edge-device budget.

8.1 Serving-Level Cache Virtualisation

Before making decisions about which tokens should be pruned out, the serving system must also ensure that its efforts in preserving tokens do not waste any cache resources on those very tokens it decides to keep. Contiguous cache allocation does not fit the autoregressive serving paradigm well because the length of the incoming requests increases dynamically, stops at some point in time, and usually shares prefixes. The idea behind PagedAttention (Kwon et al., 2023) is that in order to overcome such limitation of the serving infrastructure, KV caches could behave more similarly to virtual memory: the caches for different requests can be partitioned into blocks of equal sizes and allocated non-contiguously, allowing dynamic allocation, prefix sharing, and copy-on-write mechanisms in decoding branches. This way, the semantics of cache contents remain intact while fragmentation and duplication of data in cache are avoided.

8.2 Streaming Retention and Heavy-Hitter Eviction

Having dealt with efficient memory allocation, another issue is that token age is a bad measure for token importance. While the naive approach uses the sliding window which always keeps the latest state while removing the oldest one, streaming generation becomes difficult when the early position of tokens functions as an *attention sink*. The combination of keeping both the initial attention sink tokens and the sliding window helps improve streaming performance without tuning (Xiao et al., 2024). This is still not enough, because tokens that have historical importance do not necessarily come from the front part of the prompt. The heavy hitter eviction approach keeps the small number of historical tokens that attract large amounts of attention mass along with the sliding window (Zhang et al., 2023). This leads us to a more accurate principle: the KV eviction should focus on attentional structure, not only recency.

That principle becomes fragile in multimodal reasoning. Multi-hop benchmarks such as HotpotQA, MuSiQue, and 2WikiMultiHopQA require evidence to be composed across separated facts (Yang et al.,

2018; Trivedi et al., 2022; Ho et al., 2020); the analogous MLLM case may involve a small visual region, an OCR token, and a distant text query. Such evidence can be low-frequency or temporarily low-attention before it becomes decisive. Thus the same attention-based heuristics that reduce cache size also create a failure mode: latent evidence may be deleted before the model has reached the reasoning step that requires it.

8.3 Prompt-Aware and Layerwise Cache Compression

Global eviction rules also ignore that attention behaviour varies across heads and layers. If a short observation window near the end of the prompt already reveals which positions a head will rely on during generation, then compression can be made prompt-aware rather than static. SnapKV operationalises this observation by clustering and retaining important KV positions per attention head, reducing long-context memory and decoding cost without fine-tuning (Li et al., 2024). Yet head-level selection still allocates cache uniformly across depth, even though lower layers tend to distribute attention broadly while upper layers funnel information into fewer critical positions. PyramidKV turns this layerwise asymmetry into a cache schedule, retaining more entries in lower layers and fewer in upper layers (Cai et al., 2025).

The other weakness is that of granularity. Token-based eviction may retain plenty of individual salient states while breaking the semantic units they form. This is especially problematic when considering MLLMs, where the semantics of visual patches, OCR chunks, subtitles, and dialogues are derived collectively rather than on an individual token basis. ChunkKV uses semantic chunks rather than individual tokens as the basis for compression (Liu et al., 2025b). In our efficiency discussion, this change is significant due to its impact on retaining the information structure rather than reducing the number of tokens.

8.4 Graph-Based and Hybrid Eviction

Chunkwise retention is still prone to missing dependencies which are not local within the flattened sequence. A visual patch, an OCR span, and a token from a question may be distantly separated within the stream but highly dependent as evidence. Graph-based eviction solves this problem by modelling the dependency structure between different tokens and evicting those that are replaceable considering the dependency graph (Li et al., 2025c). Thus, the use of a graph representation is not just a more advanced scoring scheme, but corresponds to the evidence dependency induced by multimodal prompts.

This same problem arises in optimisation of different stages. For example, if upstream compression has already reduced the visual tokens by eliminating redundancy, then the remaining visual KV entries come from a completely different distribution compared to a naive decompressed prompt. Downstream cache management with a fixed eviction strategy will over-evict the very tokens that were kept due to informative reasons. Sparsification and cache compression that coordinates across stages addresses this mismatch by distributing compression across attention heads, accelerating prefilling by 2.3× and improving decoding throughput by 2.8× with only a 0.5% VQAv2 accuracy reduction (He et al., 2024).

8.5 Cross-Modal and State-Space Cache Compression

The largest gap exists between the two modalities themselves. While tokens from text usually possess logical structure, tokens in the image modality might be based on distributed perceptual evidence; applying the same eviction condition to both results in the over-retention of repeated evidence from images, but loss of rare information needed for reasoning tasks. Modality-adaptive eviction resolves this issue by using a different criterion for each modality, resulting in a 1.3×–1.5× reduction in decoding latency (Li et al., 2025b). Consequently, modality-based caching becomes a multimodal resource allocation problem, where allocating resources depends not only on the amount of attention paid, but its form.

Finally, if exact KV storage remains too expensive even after modality-aware eviction, the cache need not be represented entirely as discrete key-value states. A compact state-space memory can absorb part of the history, while entropy-based eviction preserves entries with the most unique information. This hybrid cache-memory design achieves 5× KV compression with 1.5× decoding acceleration on MLLM benchmarks (Liu et al., 2026a). The broader implication is that KV-cache optimisation is no longer a systems afterthought.

It is a modelling decision that determines which parts of a multimodal context remain addressable during generation, and therefore which reasoning chains can survive under edge-memory constraints.

The reported speedups and compression ratios throughout this section—2.3× prefilling acceleration with 2.8× decoding-throughput improvement at a 0.5% VQAv2 loss (He et al., 2024), 1.3×–1.5× decoding gains (Li et al., 2025b), and 5× compression with 1.5× acceleration (Liu et al., 2026a)—are typically measured as small benchmark-level losses or accuracy parity on the benchmarks used to evaluate the method, most commonly single-hop QA and standard long-context perplexity. This is a narrower claim than "no information loss." The multi-hop evidence-deletion failure mode described above operates on low-frequency cached states that may not be exercised by these benchmarks at all, so a method can report accuracy parity while still discarding entries that would matter for a held-out multi-hop or fine-grained visual query. The accuracy cost of KV-cache efficiency is therefore best understood as conditional on the evaluation distribution: aggressive eviction is cheap in accuracy terms for the queries a benchmark happens to ask, and potentially expensive for the queries it does not.

9 Mixture-of-Experts: Architectures and Routing

Mixture-of-Experts architectures improve scaling by activating only a small subset of expert feed-forward networks for each token. Their appeal is that sparse activation can increase model capacity without increasing per-token arithmetic in the same proportion, making MoE models attractive for efficient multimodal inference. The edge failure case is that routing is not merely a mathematical selection rule: each expert assignment becomes a memory-access and scheduling decision, and shifted token distributions can produce load imbalance, expert collapse, or unstable specialisation. This section therefore examines routing not only as an accuracy mechanism, but as the control surface that determines whether sparse capacity remains usable under compression, quantization, and hardware constraints.

9.1 The Routing Design Space: Differentiability, Balance, and Collapse Avoidance

The main difficulty with MoE routing is balancing three somewhat conflicting goals, namely *differentiability* (gradients must flow through the routing layer in order to facilitate proper specialisation of the experts), *load balance* (all experts need to be roughly equally active in terms of token stream traffic to avoid inefficiency), and *non-collapse* (the router cannot fall into the habit of choosing just a few experts all the time). The classical Top-k router with additional balancing losses achieves the latter goal using regularisation, but fails in differentiability due to the discontinuity introduced by the argmax operation.

Differentiability, balancing, and collapse avoidance are achieved using different approaches; however, different risks remain even after achieving such improvements. First, Continuous ReLU Routing replaces the hard Top-k argmax approach to allow backpropagation while still needing proper handling to avoid relaxation routing problems (Wang et al., 2025c). Second, load balancing approaches solve the problem without introducing auxiliary losses and tuning their weights but fail to address the differentiability issue (Wang et al., 2024a). Third, eigenvector routing eliminates the problem of collapsing by mapping the input tokens to the eigenvectors of the expert matrices in a way that guarantees non-collapse assignments without changing the training process. (Do et al., 2026). A unified routing system would therefore combine continuous optimisation, structural load balance, and collapse-resistant expert geometry rather than treating them as separate fixes.

9.2 Expert Collapse: Quantitative Diagnosis and Failure Mode Taxonomy

EXpert collapse—the router continuously selects a small number of experts while underutilising other experts—is the key failure mode for training MoEs. It is critical to understand what collapse means in terms of defining its mitigation mechanisms, since "collapse" can refer to different concepts with varying thresholds and implications. An expert is considered collapsed when its *utilisation rate* becomes less than $1/N$ of the allocation to uniform distribution across experts, with N being the total number of experts. In a model with 64 experts, this threshold amounts to 1.5625%; thus, any expert with less than this utilisation rate makes an ineffective contribution and can be seen as wasted capacity. Empirically, collapse

can spread across depth, with roughly one-third of sparse MoE layers drifting toward single-expert routing when balancing is weak (Sha, 2026). When it comes to conditional generation, there appears another form of collapse—*selective deadlock*, which happens when tokens of a certain class are routed to a single expert leading to its capacity overflow and 8%–15% loss in class-conditioned FID scores (Sha, 2026).Since post hoc utilisation statistics identify failure only when harm occurs, spectral characteristics of the route probability distribution become a more preventive measure: a high routing probability Gini coefficient (i.e., above 0.7) indicates network collapse, regardless of the topology (Delibasoglu, 2026).

Hierarchical expert architecture provides a scaling solution to the problem: gate networks based on task specialization create soft limits that enhance load balancing and specialized task performance, thus ensuring that no expert takes control over all tasks (Wang et al., 2024b).

9.3 Representation Collapse in Tokenisation

Analogous collapse behaviors emerge within vector quantization (VQ) architectures, where codebook collapse refers to the phenomenon wherein a few codebook elements become the exclusive choice of the model while the rest remain unused. Codebook collapse exhibits a rich-get-richer behavior, whereby codebook elements with assignment probabilities greater than the mean assignment probability experience norm growth, receive an increasing number of assignments, and thereby further contribute to the disparity (Zhao et al., 2024). The use of a simple linear layer on codebook elements may counteract such a feedback mechanism by ensuring linear independence among codebook elements (Zhu et al., 2025). Nevertheless, this static solution does not carry over into the domain of MoE since the MoE utilization problem involves determining a probability distribution dynamically over expert weight matrices based on the input token mini-batch; unlike the fixed codebook structure of VQ, no combination of expert weight matrices and the token mini-batch can produce an orthogonal or linearly independent set of experts.

9.4 MoE in Vision and Multimodal Models

Whether MoE routing benefits visual tasks is dependent on the computational regime. Sparse routing is optimal when the dense backbone is inadequate to model all token types, such that specialized experts complement the backbone representation; otherwise, memory access and load imbalance due to routing can become dominant (Sun et al., 2026). This is particularly relevant for compression, where strongly compressed architectures, due to their smaller backbone capacity, stand to benefit more from routing compared to uncompressed networks. CosMoES proposes compact sparse architectures to minimize MoE routing overhead when operating with few parameters (Huber et al., 2025), and lookup-based experts, which share an embedding table to drastically lower expert parameter costs (Jie et al., 2025).

10 High-Throughput MoE Serving

High-performance MoE serving turns routing into a systems problem. Sparse models become efficient if the set of chosen experts can be efficiently loaded, cached, and executed without spending too much effort on memory stall or cache eviction. It is particularly important for edge and near-edge execution where the weights of the experts are very heavy, expert traffic varies based on inputs, and the performance of the model depends as much on data accesses as on FLOPS performed. The failure mode is that while conceptually sparse, such a model could run inefficiently due to being memory bound because of uncertain routing, imbalance in the number of requests to different experts, and competition of the attention and execution cache spaces.

10.1 System-Level Analysis and Scheduling

The throughput limit for high-throughput MoE serving is not constrained by any one bottleneck, but rather by the point at which the task shifts from compute-bound to memory bandwidth-bound. Such a model can be used to optimize batching, scheduling, and offloading, providing speedup of up to 25.5× (4.6× on average) relative to an unsophisticated dense baseline model. (Yuan et al., 2025).

Memory stalls can be reduced either by predicting routing paths early enough to prefetch required experts, or by continuously adapting the prefetch policy as workload statistics change. Predictive routing paths hide memory latency and can yield large GPU memory savings with 2–10× inference speedups (He et al., 2024); adaptive prefetching then reduces residual stall time to near-zero levels under changing loads (Shen et al., 2025b).

Serving efficiency can also be improved before memory management begins. Sensitivity- based gating skips low-sensitivity experts for compatible token types, reducing the average number of activated experts by 25% and achieving a 1.35× speedup without memory-management overhead (Zhong et al., 2024). When workload heterogeneity is the dominant issue, task-aware expert loading adapts memory allocation to request characteristics and improves utilisation across diverse tasks (Tairin et al., 2025).

10.2 Disaggregated Serving

Co-locating attention and expert modules creates cache interference because attention benefits from full KV residency while experts benefit from parallel weight movement. Disaggregating the two module types onto separate GPU sub-clusters preserves fast KV access for attention and lets expert execution proceed without cache interference, achieving up to 3.9× higher per-GPU throughput than co-located serving (Zhang et al., 2026).

11 Quantization for Efficient Inference

Quantization lowers the precision of weights and activations so that models take up less space in memory, transfer less data, and are cheaper to compute. It is crucial for edge deployment when bandwidth and storage are the dominating factors in terms of energy consumption, meaning that lowering precision can yield advantages that are not achievable by just reducing FLOPs. However, with multimodal and MoE models, quantization adds new failure modes: extreme activation values cause computing to fail; router logits get corrupted and experts change; and vision and language models can accept different precisions. Table 4 condenses the main quantization failure regimes discussed in this section and makes explicit that not all failures are recoverable by the same kind of calibration strategy.

11.1 Quantization Failure Modes: A Quantitative Taxonomy

Before exploring quantization techniques separately, we first need to understand their failure mechanisms using evidence that the cited papers actually report. One useful distinction separates gradual signal degradation from catastrophic computation collapse, because the two modes require different remedies (Zhou et al., 2026). Signal degradation is the regime in which quantization perturbs computation while preserving the overall processing pattern, whereas computation collapse is the regime in which key operations fail and the forward pass breaks down. For LLaMA-style transformers, Zhou et al. (2026) show that these regimes separate cleanly in practice: 4-bit quantization is usually the workable trade-off, while pushing to 2-bit typically triggers a catastrophic performance cliff. The practical significance of this taxonomy is that gradual degradation can often be mitigated by calibration or lightweight fine-tuning, whereas collapse usually requires architectural intervention such as clipping, mixed precision, or structural repair.

MoE quantization adds a third failure mode: *routing inconsistency*, where noise in router logits changes expert assignments post-deployment. Rather than claiming a universal assignment-change threshold, the clearest source-backed evidence is outcome-based: routing-aware post-training quantization recovers 1.15%–2.28% average score across three MoE architectures under W4A4 and W3A4 settings, implying that preserving routing structure matters empirically even when aggregate quantization error appears modest (Fu et al., 2026). Figure 6 recasts these failure modes as a side-by-side comparison of three regimes. Rather than presenting a decision path, the figure highlights what breaks in each case, what evidence marks the regime, and which mitigation is most appropriate, making clear that quantization can perturb signal fidelity, sparse routing, or numerical stability at different levels of severity.

Table 4: Taxonomy of quantization failure modes in VLMs and MoE models. Rec. = recoverable via calibration/fine-tuning without architectural changes (∼ = partial).

Failure mode	Representative evidence	Rec.	Mitigation	Scope
Signal degradation (Zhou et al., 2026)	Gradual degradation in the 4-bit regime	✓	Calibration or fine-tuning	VLM, MoE
Computation collapse (Zhou et al., 2026)	Catastrophic cliff in the 2-bit regime	✗	Mixed precision / clipping / structural repair	VLM, MoE
Routing inconsistency (Fu et al., 2026)	Routing-aware PTQ recovers 1.15%–2.28% average score at W4A4/W3A4	∼	Calibration- (Fu et al., 2026) or constraint-centric (Park et al., 2026)	MoE only
Modality-heterogeneous degradation (Zhong et al., 2026)	SplitQ retains 93.5% of FP16 at W3A3 (69.5 vs. 74.3)	∼	Modality-separated quantization (Xue et al., 2026)	VLM only
Speculative decoding interaction (Zhang et al., 2025c)	Hierarchical design reaches 2.78× speedup and 1.31× over EAGLE-2 on 4-bit Llama-3-70B	✓	Joint drafter–target design	VLM, MoE

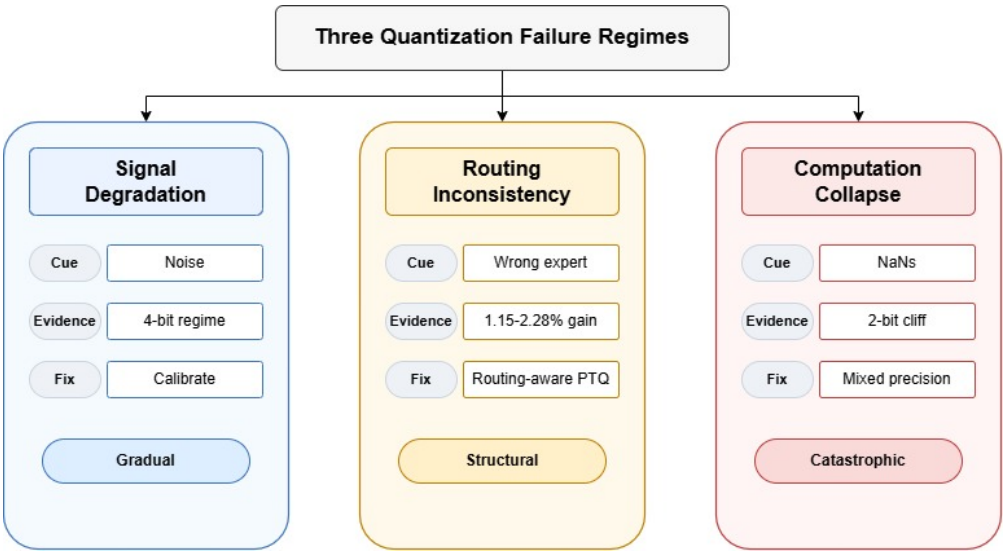

Figure 6: Comparison of three quantization failure regimes in VLM and MoE inference. Signal degradation is the gradual regime in which outputs remain valid but noisier and can often be mitigated by calibration. Routing inconsistency is the structural MoE-specific regime in which quantized router logits perturb expert selection; the evidence shown in the figure reflects outcome-based recovery from routing-aware post-training quantization (1.15%–2.28% average score under W4A4/W3A4 settings). Computation collapse is the catastrophic regime in which low-bit quantization breaks numerical stability, motivating mixed-precision or other architectural repair.

11.2 Addressing Routing Inconsistency: Structure-Preserving Quantization

Two complementary strategies address routing inconsistency. Calibration-centric quantization exposes the router to token distributions that most frequently produce assignment changes and then minimises logit perturbation on those distributions (Fu et al., 2026). Constraint-centric quantization instead preserves the full-

precision Top-k assignment structure on calibration data, penalising any quantized parameter configuration that changes expert assignments (Park et al., 2026). The first strategy reduces average perturbation; the second eliminates assignment changes that would remain despite reduced perturbation, with no inference-time overhead.

11.3 Global and Mixed-Precision Quantization for MoE

Mixed-precision MoE quantization shifts the allocation unit from layers to experts. Global expert-importance scoring assigns higher precision to rare but critical experts and aggressively compresses routine experts, substantially reducing memory demand while maintaining accuracy (Deng et al., 2026). Low-rank compensation complements this allocation by learning fine-grained corrections for the most critical experts, improving the bandwidth-accuracy trade-off of offloaded MoE inference (Liu et al., 2025d).

At extremely low precision, the central trade-off becomes representation expressivity versus bitwise efficiency. Experts using binary models compress weights into $\{-1, +1\}$ with learned compensation for accuracy restoration along with bitwise computation kernels (Zhao et al., 2026). Experts using ternary models incorporate zero in the activation space which provides an architectural advantage because it matches the sparse activation pattern, unlike generic bit width choices (Wang et al., 2026). Scaling law analysis helps understand the cases where binary or ternary architectures are preferable: under 4-bit quantization, the Pareto frontier switches from calibration of the architecture to architecture selection (Liu et al., 2025c).

11.4 Modality-Heterogeneous Quantization in VLMs

The most distinct structural problem in VLM quantization is the presence of visual and textual tokens with vastly different activation distributions. Visual tokens have higher activation variance between tokens (visual token std is up to 4× higher than text tokens in LLaVA-class models), more outliers (outlier probability $\sim 2.3\%$ for visual tokens vs. $\sim 0.6\%$ for text tokens with similar architectures), and activation sensitivities in different weight dimensions. Naively quantizing visual and textual tokens jointly at W4A8 leads to 2%–4% accuracy loss in VQA compared to modality-separated quantization, with accuracy losses concentrated in spatial reasoning tasks where visual token precision is important (Zhong et al., 2026). Second order curvature information is useful for identifying weight dimensions that are sensitive to quantization noise, and applying high-precision quantization there (Xue et al., 2026). Binary quantization of VLM models and even 2-bit VLM models need architecture changes beyond post-training calibration, since calibration alone is not enough (Wang et al., 2025b; Guo et al., 2025).

11.5 Cross-Technique Interactions: Speculative Decoding and Quantization

Quantization modifies the distribution of the target model, leading to a higher likelihood of rejection of the drafts generated by the drafter. At W8A8, the acceptance ratio falls by 3%–5% relative to full precision, mitigating the gains in performance from speculative decoding by 1.5%–2×. Calibration of the drafter-target couple, along with the quantized target, can help compensate for this interplay loss (Zhang et al., 2025c). This interaction is a further instance of the Interface C pattern from Section 3: a precision change applied to the target model propagates into a separate component (the drafter) whose calibration assumptions were not designed to account for it, producing a system-level efficiency loss that single-component evaluation would miss.

12 Edge and On-Device Deployment

However, edge/deployment in device is when the algorithms that make the survey efficient encounter hardware limitations. While a model is suitable for running on a server-side GPU, it might not work well on a mobile or embedded device due to constraints such as SRAM, bandwidth, latency for transferring data, power, and thermal throttling. In MoE models, the main limitation is the fact that expert weights are much larger than fast on-chip memory and, therefore, dynamic loading from slower memory happens after every routing call. These methods help in reducing this problem, but they tend to disrupt routing locality and caching.

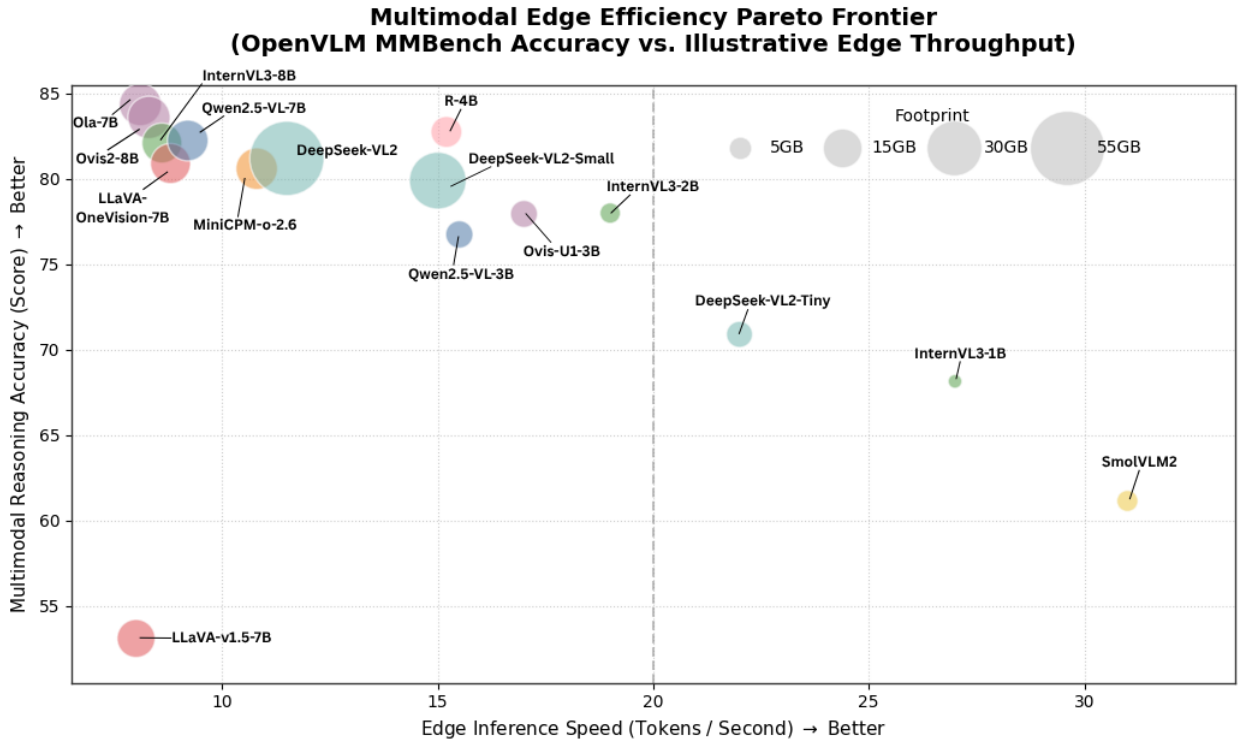

Figure 7: Illustrative multimodal edge-efficiency Pareto frontier. Accuracy is represented by OpenVLM MMBench V1.1 score, indicative edge throughput is shown on the horizontal axis, and bubble size denotes approximate FP16 model footprint. The figure highlights an edge deployment failure mode: high-accuracy models remain memory- and latency-heavy, while models optimized for faster edge inference often sacrifice reasoning accuracy. The empty upper-right region captures the difficulty of achieving high accuracy, high throughput, and low memory footprint simultaneously.

12.1 Expert Storage Hierarchies

Hot-cold hierarchy controls the placement of experts within the device itself. Layers that do not require experts to be specialized, including embedding, attention, and layer normalization, are retained in the DRAM, whereas the weights of expert FFN are considered cold and prefetched into the memory based on routing predictions due to expert selection locality. This yields substantial speedups over naive all-in-storage baselines (Yi et al., 2025b).

Prefetch predictors and tiered storage strategies can increase the depth of the hierarchy to support larger sparsity-based models on mobile devices (Yi et al., 2025a).

Offloading techniques can be complemented by precision-aware nested representations; when weights are stored in a quantized manner, experts can be accessed at varying degrees of precision based on their usage, thereby saving memory space while not being limited to one level of precision (Wang et al., 2025a).

12.2 Lossless Compression and Scheduling

Lossless expert-weight compression and cache-aware scheduling tackle two separate aspects of the device-side inference latency chain. The former aims to lower transfer costs, while the latter maximizes cache hits through allocation of commonly triggered experts to adjacent cache addresses, resulting in an up to 72.77% reduction in inference latency compared to the unoptimized MoE serving process (Yang et al., 2026).

Heterogeneous expert quantization also tightly ties model architecture and hardware capabilities: heterogeneous quantization allows assigning different precision to various experts based on importance and edge device limitations, thus achieving a better accuracy vs deployment cost Pareto curve. (Liu et al., 2026b).

Most of the efficiency measures mentioned earlier–throughput, decoding time latency, transfer volume per token–presumably assume the existence of a consistent inference load where the model is loaded once and runs many queries, so that there is hardly any loading cost at all. This assumption does not hold in many edge/mobile use cases where the model is loaded sporadically, and the device clears its memory between loads in order to allocate space for other processes, such that each invocation carries a cold-start overhead of loading expert models and backbone from storage before generating the first token. For MoE models where the experts are hot-cold and pre-fetched (Yi et al., 2025b,a), In short interactions, the start-up overhead can outweigh the end-to-end latency, even though the cost in the steady state is small per token. A compressor/quantizer that achieves low steady-state transfer cost need not also achieve low total latency for an irregularly accessed use-case, since lowering the precision leads to lower transfer cost for loading bytes prior to inference. Measuring the loading time and the inference latency separately, and assessing both with an appropriate intermittent-use workload, is thus important for making edge deployment claims representative; this aspect of loading latency has been overlooked in most of the throughput-centric benchmarks summarized in Section 13. The very same memory hierarchy that influences the steady-state cost for loading an expert dictates how a device can host all of the experts or needs to coordinate with other devices, and this is one motivation behind the distributed/federated deployment scenario considered in Section 14.

Figure 7 makes the edge-deployment trade-off visually explicit. High-accuracy models cluster toward the upper-left region, but their larger memory footprint and lower indicative throughput make direct on-device deployment difficult. Models that move toward the high-throughput regime tend to occupy lower accuracy regions, and the sparsely populated upper-right corner captures the current "edge hardware wall" where high accuracy, high throughput, and low memory footprint remain difficult to achieve simultaneously.

13 Benchmarking and Evaluation Frameworks

Benchmarking determines whether claimed efficiency survives contact with real models, real prompts, and real hardware. This matters because compression, quantization, routing, cache management, and edge scheduling can each improve one metric while degrading another. The current failure condition is evaluative fragmentation: existing suites measure accuracy, latency, memory, power, routing cost, or localisation in isolation, but rarely expose how an upstream optimisation creates downstream reasoning or deployment failure. This section therefore separates three questions: what current benchmarks can measure, what they cannot measure, and what a unified multimodal edge benchmark should report.

Table 5 makes the evaluative gap concrete: existing suites measure important slices of the problem, but none of them jointly stress compression, routing, KV-cache behaviour, quantization, and hardware under a single deployment protocol.

13.1 What Current Benchmarks Can Measure

Current benchmarks are strongest when the measurement target is isolated. Accelerator- level suites standardise throughput, latency, and memory utilisation across AI hardware (Chitty-Venkata et al., 2024); mobile-focused suites evaluate compressed-model accuracy-efficiency trade-offs on device classes where memory and thermal limits dominate (Li et al., 2025d); hardware-aware architecture benchmarks support NAS comparisons across simulated configurations (Sukthanker et al., 2024); and power-performance characterisation captures accelerator efficiency beyond latency alone (John et al., 2024). On the model-quality side, VQA and OpenVLM-style leaderboards measure multimodal reasoning accuracy, while localisation-specific benchmarks reveal failures hidden by aggregate VQA scores (Chiang et al., 2024). For sparse models, cost-accuracy- performance evaluation makes routing and expert movement visible rather than treating MoE systems as ordinary dense models (Jiang et al., 2025).

Table 5: Current benchmark coverage for efficient multimodal inference. Existing benchmarks provide useful single-axis measurements, but no benchmark jointly measures compression, routing, quantization, cache residency, and hardware behaviour under one deployment protocol.

Benchmark	**What it measures**	**Remaining gap**
ImageNet	Image classification accuracy and throughput	Weak proxy for multimodal reasoning
DeiT / MAE / CLIP eval suites	ViT representation quality under token merging	Limited downstream MLLM coverage
VQA benchmarks	Answer accuracy under visual-language tasks	Often miss spatial and OCR degradation
Mixtral-class reasoning benchmarks	Language reasoning after MoE quantization	Do not expose visual-token routing shift
LLaVA-class eval suites	VLM accuracy and activation statistics	Limited hardware and energy reporting
OpenVLM MMBench V1.1	Multimodal reasoning accuracy	No standard edge throughput protocol
LLMInferenceBench	Accelerator latency, throughput, and memory	Mostly hardware-centric, not multimodal-failure-centric
PalmBench	Mobile compressed-model accuracy-efficiency tradeoffs	Limited MoE and VLM interaction coverage
HW-GPT-Bench	Hardware-aware architecture-search comparisons	Simulated design space, limited deployment traces
CarAML	Power-performance characterisation	Not specialised to MLLM pipelines
MLPerf Power	Standardised power measurement	Energy axis only, not reasoning failure diagnosis
MoE-CAP	MoE cost-accuracy-performance trade-offs	Does not include visual compression or modality-aware cache effects
LocateBench	Spatial localisation robustness	Capability-specific, not full deployment-system evaluation

13.2 What Current Benchmarks Cannot Measure

What's missing here isn't another one-dimensional measurement, but rather a measurement framework which takes into account the propagation of failure across steps. Standard VQA score won't reveal any impact on localizability due to compression because the former can preserve the correctness of answers despite the spatial grounding being degraded (Chiang et al., 2024). Throughput benchmark can record an improvement in performance without showing whether it occurred due to the omission of visual tokens, a disruption of MoE routing, and/or evictions of KV pairs required for multiple hops of reasoning. Power benchmark can assess the energy consumption from *micro*-watts of embedded systems to mega-watts of data centers (Tschand et al., 2025), yet it doesn't indicate the source of energy efficiency achieved: was it due to lower precision, reduced number of visual tokens, improved KV pairs residency, or different routing of experts? The same applies to the expert-centric MoE metrics (Jiang et al., 2025) which show expert movements and their execution cost, yet not their routing or reassignment.

This matters because energy and latency are shaped by memory movement as much as by arithmetic. Energy diagnosis shows that memory bandwidth, not compute, dominates memory-bound inference (Chung et al., 2026), so a benchmark that reports FLOP reduction alone can reward optimisations that do not reduce wall-clock energy. Because inference energy can scale sharply with accuracy targets (Yang et al., 2024), accuracy-energy engineering must include sustained throughput, thermal behaviour, memory traffic, and answer quality under one protocol. Existing tools cover pieces of this space—deadline-aware frequency scaling (Yan et al., 2024), training-side power measurement (Latif et al., 2025), kernel-level NAS (La & Ha, 2025), and energy-latency-accuracy trilemma formulations (Tian et al., 2024)—but they do not jointly diagnose multimodal compression-routing-quantization failures.

13.3 Toward a Unified Multimodal Edge Benchmark

An adequate benchmark for the systems surveyed here should evaluate the full deployment tuple rather than a single number:

$$\big(\text{accuracy},\ \text{latency},\ \text{energy},\ \text{peak memory}, \text{KV footprint},\ \text{expert traffic},\ \text{routing stability}\big).$$

The benchmark must define hardware targets, prompt sizes, vision-tokens budget, precision format, and decoding method, and then measure the effect of each optimization on both system and reasoning performance. For dense vision-language models, this entails determining whether the small model size and limited number of high-quality visual tokens lead to better performance than the larger models under the same amount of compute resources (Li et al., 2025a). For the MoE vision-language models, this involves determining expert usage, expert transfer volume, routing coherence before and after quantization, and cache hit rate under vision-token compression.

The suggested benchmark should also feature stress tests aimed at activating the interfaces discussed here, namely pre-routing compression, post-routing quantization, KV-cache stress under long multimodal prompts, and sustained edge inference within the thermal envelope. For vision-language models using video MoEs, the routing stability component of the deployment tuple must explicitly include *Temporal Routing Consistency* (TRC, Section 3.5) for the video input before and after compression and quantization, to enable assessment of expert jitter between adjacent frames in parallel with the existing static FLOPs vs. accuracy tradeoffs that benchmarks currently measure. Evaluation of performance scalability through probing allows comparing capacities across model sizes and architectures (Castro et al., 2023), and security analysis adds resistance to OOD or backdoor visual inputs (Lyu et al., 2025). What is important is that efficiency improvements have to be reported together with any failure modes they may introduce.

14 Open Challenges and Future Directions

In earlier sections we already demonstrated that faster kernels and stronger compressors alone do not solve remaining hardest problems; there are no guarantees of operation at the interface between compression, routing, quantisation, cache management and hardware execution. With edge deployments requiring all of these interfaces to meet tight constraints on memory usage; latency and energy consumption, there is little chance for post-hoc correction of an error once one occurs.

The major mode of failure in these scenarios is when efficiency improvements made locally by one interface remove required evidence for operation from an interface, destabilise expert systems; change the operation of precision-dependent computations at another interface or require more movement of data from memory to the respective pipelines downward through the various interfaces. So the focus of future work in these areas will be on robust measurement from the respective stages of operations within an interface rather than simply improving the individual components of the system.

Compression-Routing Interaction in MoE. If token compression is used prior to MoE routing, the compressed tokens will create a distribution shift compared to the original tokens, possibly interfering with the learning of expert specialization. Even the comparatively mild perturbation introduced by quantization is already important enough that routing-aware post-training quantization recovers 1.15%–2.28% average score in low-bit MoE settings, so token compression, which alters representations more directly, is a plausible source of even larger routing instability. Although token compression and MoE routing interact in a way that is critical for architectures that use both methods, there have been few studies on this interaction, and future research should explore it further (Fu et al., 2026).

Temporal Routing Continuity for Video MoE. As we saw in Section 7, the lack of recurrent state in transformers for video implies that temporally continuous representation is not enforced by design, and relies on emergent expertise instead. A promising avenue of research is enforcing temporally consistent routing among neighboring tokens—either by using routing consistency losses to ensure high TRC between subsequent frames, or designing models to incorporate compact temporal information in their routing decisions, in a manner reminiscent of LSTMs.

Unified Cross-Modal Compression. Current token compression methods largely treat visual and textual modalities independently. The modality heterogeneity challenge identified by (Zhong et al., 2026) and (Li et al., 2025b) points toward modality-specific strategies, but a principled framework for joint cross-modal compression—accounting for complementary inter-modal information—remains open.

Theoretical Foundations of Expert Collapse. While multiple empirical mitigation strategies have been proposed (Wang et al., 2024a; Do et al., 2026; Delibasoglu, 2026), and practical empirical markers such as utilisation below $1/N$ and layerwise drift toward single-expert routing are beginning to emerge, a theoretical characterisation of when and why collapse occurs is incomplete. Analogies from vector quantization collapse (Zhao et al., 2024; Zhu et al., 2025) are informative but do not directly transfer to continuous-routing MoE contexts.

Proactive Failure Mode Detection. The quantization failure modes identified by (Zhou et al., 2026) and routing collapse modes by (Sha, 2026) are currently detected only post-deployment. The spectral Gini threshold (Delibasoglu, 2026) together with activation-level monitoring for collapse-prone low-bit layers (Zhou et al., 2026) represent first steps toward proactive monitoring, but neither is integrated into standard training pipelines. Systematic incorporation of collapse and degradation monitors into training dashboards, with automatic corrective interventions, represents a practically important and theoretically interesting challenge.

Long-Video Efficiency Benchmarks. Long-form video understanding (Zou et al., 2024; Faure et al., 2025) lacks benchmarks that jointly measure temporal reasoning quality and computational efficiency. Existing benchmarks reward accuracy without penalising compute cost; future benchmarks should reward high-quality answers at low inference cost.

Hardware-Aware Co-Design. The energy efficiency literature (Chung et al., 2026; Yang et al., 2024; Tschand et al., 2025) reveals that memory bandwidth—not compute—is the dominant energy consumer in memory-bound inference. This implies that quantization (which reduces bandwidth) has disproportionate energy impact relative to compute pruning (which reduces FLOPs), yet most compression research optimises FLOPs. A co-design methodology jointly optimising architecture, quantization strategy, and hardware configuration—guided by measured energy consumption per bandwidth byte rather than per FLOP—represents a significant unrealised opportunity.

Federated and Distributed Edge Inference. As noted in Section 12, the same memory hierarchy that governs on-device expert storage also determines when a single device cannot hold a model locally and must coordinate with others. Distributed inference across multiple constrained collaborating devices introduces additional load balancing, communication overhead, and privacy preservation challenges that require new frameworks (Yi et al., 2025b,a; Wang et al., 2025a).

Hallucination and Factual Accuracy Under Compression. Token compression, quantization, and KV-cache eviction have been assessed primarily by accuracy, latency, and memory metrics in this survey, yet deleting or modifying visual input and cached context can also influence what the model *says* about itself regardless of any changes in benchmark performance. While a model that no longer has the ability to see deleted visual input or reference the evicted KV buffer may choose to say "I don't know," it might instead formulate a coherent response based on contextual input without evidence. The impact of extreme compression, quantization, or eviction on the rate of hallucination generation and whether such instances are skewed toward infrequent but informationally dense tokens or experts like those that failed at Interfaces B and C has yet to be fully characterized.

15 Conclusion

This survey has argued that efficient multimodal intelligence is fundamentally a co-design problem. Visual token compression, KV-cache management, MoE routing, quantization, expert storage, and hardware execution each offer meaningful efficiency gains, but their effects are coupled across the inference pipeline. A compression decision can remove visual evidence or shift router inputs, a routing decision can alter quantization sensitivity and expert movement, a cache eviction policy can erase information needed later for

reasoning, and an edge memory or energy constraint can overturn what appears to be an efficient model-level optimisation.

The main lesson is that efficient multimodal systems must be designed around interfaces, not isolated components. Preserving information through compression, maintaining routing stability under shifted token distributions, calibrating quantization to actual expert usage, and retaining the right cached evidence are all part of the same deployment problem. The failure-chain taxonomy developed in this survey makes these dependencies explicit and shows why local improvements in FLOPs, token count, or bit-width are insufficient unless they survive downstream routing, caching, quantization, and hardware constraints.

Future progress will require evaluation protocols that report accuracy, latency, energy, memory footprint, KV-cache usage, expert traffic, routing stability, and reasoning reliability together. For video and long-context multimodal models, diagnostics such as Temporal Routing Consistency can expose failures that static accuracy metrics miss. The next generation of efficient multimodal systems should therefore move beyond component-wise optimisation toward integrated deployment design, where models are not only small and fast, but also reliable under the resource constraints where efficient intelligence matters most.

References

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S. Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models, 2024. URL https://arxiv.org/abs/2408.10945.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu. ICLR 2023 notable top 5%.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling, 2025. URL https://arxiv.org/abs/2406.02069.

Santiago Castro, Oana Ignat, and Rada Mihalcea. Scalable performance analysis for vision-language models, 2023. URL https://arxiv.org/abs/2305.18786.

Junjie Chen, Xuyang Liu, Zichen Wen, Yiyu Wang, Siteng Huang, and Honggang Chen. Variation-aware vision token dropping for faster large vision-language models, 2026. URL https://arxiv.org/abs/2509.01552.

Ruichen Chen, Keith G. Mills, Liyao Jiang, Chao Gao, and Di Niu. Re-ttention: Ultra sparse visual generation via attention statistical reshape, 2025. URL https://arxiv.org/abs/2505.22918.

Ting-Rui Chiang, Joshua Robinson, Xinyan Velocity Yu, and Dani Yogatama. Locatebench: Evaluating the locating ability of vision language models, 2024. URL https://arxiv.org/abs/2410.19808.

Krishna Teja Chitty-Venkata, Siddhisanket Raskar, Bharat Kale, Farah Ferdaus, Aditya Tanikanti, Ken Raffenetti, Valerie Taylor, Murali Emani, and Venkatram Vishwanath. Llm-inference-bench: Inference benchmarking of large language models on ai accelerators, 2024. URL https://arxiv.org/abs/2411.00136.

Jae-Won Chung, Ruofan Wu, Jeff J. Ma, and Mosharaf Chowdhury. Where do the joules go? diagnosing inference energy consumption, 2026. URL https://arxiv.org/abs/2601.22076.

Ibrahim Delibasoglu. Spectral manifold regularization for stable and modular routing in deep moe architectures, 2026. URL https://arxiv.org/abs/2601.03889.

Jianing Deng, Song Wang, Dongwei Wang, Zijie Liu, Tianlong Chen, Huanrui Yang, and Jingtong Hu. Gemq: Global expert-level mixed-precision quantization for moe llms, 2026. URL https://arxiv.org/abs/2605.23078.

Giang Do, Hung Le, and Truyen Tran. Eigenvectors of experts are training-free non-collapsing routers, 2026. URL https://arxiv.org/abs/2605.30992.

Katrina Drozdov, Ravid Shwartz-Ziv, and Yann LeCun. Video representation learning with joint-embedding predictive architectures, 2024. URL https://arxiv.org/abs/2412.10925.

Xianzhi Du, Tom Gunter, Xiang Kong, Mark Lee, Zirui Wang, Aonan Zhang, Nan Du, and Ruoming Pang. Revisiting moe and dense speed-accuracy comparisons for llm training, 2024. URL https://arxiv.org/abs/2405.15052.

Gueter Josmy Faure, Jia-Fong Yeh, Min-Hung Chen, Hung-Ting Su, Shang-Hong Lai, and Winston H. Hsu. Hermes: temporal-coherent long-form understanding with episodes and semantics, 2025. URL https://arxiv.org/abs/2408.17443.

Zhongqian Fu, Tianyi Zhao, Ning Ding, Xianzhi Yu, Xiaosong Li, Yehui Tang, and Yunhe Wang. Eaquant: Enhancing post-training quantization for moe models via expert-aware optimization, 2026. URL https://arxiv.org/abs/2506.13329.

Tianyu Guo, Shanwei Zhao, Shiai Zhu, and Chenguang Ma. Speed-q: Staged processing with enhanced distillation towards efficient low-bit on-device vlm quantization, 2025. URL https://arxiv.org/abs/2511.08914.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/He_Masked_Autoencoders_Are_Scalable_Vision_Learners_CVPR_2022_paper.html.

Xin He, Shunkang Zhang, Kaijie Tang, Shaohuai Shi, Yuxin Wang, Zihao Zeng, Zhenheng Tang, Xiaowen Chu, Haiyan Yin, Ivor W. Tsang, and Yew Soon Ong. ExpertFlow: Efficient Mixture-of-Experts Inference via Predictive Expert Caching and Token Scheduling. *arXiv e-prints*, art. arXiv:2410.17954, October 2024. doi: 10.48550/arXiv.2410.17954.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Zipvl: Efficient large vision-language models with dynamic token sparsification, 2024. URL https://arxiv.org/abs/2410.08584.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps, 2020. URL https://arxiv.org/abs/2011.01060.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Hsiang-Wei Huang, Wenhao Chai, Kuang-Ming Chen, Cheng-Yen Yang, and Jenq-Neng Hwang. Tosa: Token merging with spatial awareness, 2025. URL https://arxiv.org/abs/2506.20066.

Mouxiao Huang, Borui Jiang, Dehua Zheng, Hailin Hu, Kai Han, and Xinghao Chen. Ppe: Positional preservation embedding for token compression in multimodal large language models, 2026. URL https://arxiv.org/abs/2510.22936.

Ruoxiang Huang and Zhen Yuan. Modix: A training-free multimodal information-driven positional index scaling for vision-language models, 2026. URL https://arxiv.org/abs/2604.12537.

Patrick Huber, Akshat Shrivastava, Ernie Chang, Chinnadhurai Sankar, Ahmed Aly, and Adithya Sagar. Cosmoes: Compact sparse mixture of experts, 2025. URL https://arxiv.org/abs/2503.00245.

Simin Huo and Ning Li. Mame & mare: Matrix-based token merging and restoration for efficient visual perception and synthesis, 2026. URL https://arxiv.org/abs/2604.13432.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4651–4664. PMLR, 2021. URL https://proceedings.mlr.press/v139/jaegle21a.html.

Yinsicheng Jiang, Yao Fu, Yeqi Huang, Ping Nie, Zhan Lu, Leyang Xue, Congjie He, Man-Kit Sit, Jilong Xue, Li Dong, Ziming Miao, Dayou Du, Tairan Xu, Kai Zou, Edoardo Ponti, and Luo Mai. Moe-cap: Benchmarking cost, accuracy and performance of sparse mixture-of-experts systems, 2025. URL https://arxiv.org/abs/2412.07067.

Shibo Jie, Yehui Tang, Kai Han, Yitong Li, Duyu Tang, Zhi-Hong Deng, and Yunhe Wang. Mixture of lookup experts, 2025. URL https://arxiv.org/abs/2503.15798.

Xinqi Jin, Hanxun Yu, Bohan Yu, Kebin Liu, Jian Liu, Keda Tao, Yixuan Pei, Huan Wang, Fan Dang, Jiangchuan Liu, and Weiqiang Wang. Streamingassistant: Efficient visual token pruning for accelerating online video understanding, 2025. URL https://arxiv.org/abs/2512.12560.

Chelsea Maria John, Stepan Nassyr, Carolin Penke, and Andreas Herten. Performance and power: Systematic evaluation of ai workloads on accelerators with caraml. In *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1164–1176. IEEE, November 2024. doi: 10.1109/scw63240.2024.00158. URL http://dx.doi.org/10.1109/SCW63240.2024.00158.

Young Jin Kim, Raffy Fahim, and Hany Hassan Awadalla. Mixture of quantized experts (moqe): Complementary effect of low-bit quantization and robustness, 2023. URL https://arxiv.org/abs/2310.02410.

Andrew Kiruluta, Preethi Raju, and Priscilla Burity. Cs-vlm: Compressed sensing attention for efficient vision-language representation learning, 2025. URL https://arxiv.org/abs/2507.02957.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th ACM Symposium on Operating Systems Principles*, pp. 611–626. ACM, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Hoang-Loc La and Phuong Hoai Ha. Kernel-level energy-efficient neural architecture search for tabular dataset, 2025. URL https://arxiv.org/abs/2504.08359.

Imran Latif, Alex C. Newkirk, Matthew R. Carbone, Arslan Munir, Yuewei Lin, Jonathan Koomey, Xi Yu, and Zhihua Dong. Single-node power demand during ai training: Measurements on an 8-gpu nvidia h100 system. *IEEE Access*, 13:61740–61747, 2025. ISSN 2169-3536. doi: 10.1109/access.2025.3554728. URL http://dx.doi.org/10.1109/ACCESS.2025.3554728.

Jaeyeon Lee and Dong-Wan Choi. *Lossless Token Merging Even Without Fine-Tuning in Vision Transformers*. IOS Press, October 2025. ISBN 9781643686318. doi: 10.3233/faia250832. URL http://dx.doi.org/10.3233/FAIA250832.

Semi Lee, Hyejin Go, and Hyesong Choi. Adamerge: Salience-aware adaptive token merging for training-free acceleration of vision transformers, 2026. URL https://arxiv.org/abs/2605.27465.

Kevin Y. Li, Sachin Goyal, Joao D. Semedo, and J. Zico Kolter. Inference optimal vlms need fewer visual tokens and more parameters, 2025a. URL https://arxiv.org/abs/2411.03312.

Kunxi Li, Zhonghua Jiang, Zhouzhou Shen, Zhaode Wang, Chengfei Lv, Shengyu Zhang, Fan Wu, and Fei Wu. Madakv: Adaptive modality-perception kv cache eviction for efficient multimodal long-context inference, 2025b. URL https://arxiv.org/abs/2506.15724.

Xuelin Li, Xiangqi Jin, and Linfeng Zhang. Graphkv: Breaking the static selection paradigm with graph-based kv cache eviction, 2025c. URL https://arxiv.org/abs/2509.00388.

Yicong Li, Xun Yang, An Zhang, Chun Feng, Xiang Wang, and Tat-Seng Chua. Redundancy-aware transformer for video question answering. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pp. 3172–3180. ACM, October 2023. doi: 10.1145/3581783.3612577. URL http://dx.doi.org/10.1145/3581783.3612577.

Yilong Li, Jingyu Liu, Hao Zhang, M Badri Narayanan, Utkarsh Sharma, Shuai Zhang, Pan Hu, Yijing Zeng, Jayaram Raghuram, and Suman Banerjee. Palmbench: A comprehensive benchmark of compressed large language models on mobile platforms, 2025d. URL https://arxiv.org/abs/2410.05315.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation, 2024. URL https://arxiv.org/abs/2404.14469.

Zizhong Li, Haopeng Zhang, and Jiawei Zhang. Mmvir: A multi-modal and multi-granularity representation for long-range video understanding, 2026. URL https://arxiv.org/abs/2601.05495.

Jiacheng Liu, Peng Tang, Wenfeng Wang, Yuhang Ren, Xiaofeng Hou, Pheng-Ann Heng, Minyi Guo, and Chao Li. A survey on inference optimization techniques for mixture of experts models, 2025a. URL https://arxiv.org/abs/2412.14219.

Sihao Liu, YuFan Xiong, Zhonghua Jiang, Zhaode Wang, and chengfei lv Shengyu Zhang. Retentivekv: State-space memory for uncertainty-aware multimodal kv cache eviction, 2026a. URL https://arxiv.org/abs/2605.04075.

Tao Liu, Qipeng Xie, Zhaoyang Han, Weizheng Wang, Shengli Pan, Xiang Zhang, Chunhua Su, and Kaishun Wu. Efficient edge deployment of mixture-of-experts models with hybrid expert quantization. *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2026b. doi: 10.1109/TCCN.2026.3694872.

Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. Multi-stage vision token dropping: Towards efficient multimodal large language model, 2024. URL https://arxiv.org/abs/2411.10803.

Xiang Liu, Zhenheng Tang, Peijie Dong, Zeyu Li, Yue Liu, Bo Li, Xuming Hu, and Xiaowen Chu. Chunkkv: Semantic-preserving kv cache compression for efficient long-context llm inference, 2025b. URL https://arxiv.org/abs/2502.00299.

Zechun Liu, Changsheng Zhao, Hanxian Huang, Sijia Chen, Jing Zhang, Jiawei Zhao, Scott Roy, Lisa Jin, Yunyang Xiong, Yangyang Shi, Lin Xiao, Yuandong Tian, Bilge Soran, Raghuraman Krishnamoorthi, Tijmen Blankevoort, and Vikas Chandra. Paretoq: Improving scaling laws in extremely low-bit llm quantization, 2025c. URL https://arxiv.org/abs/2502.02631.

Zhenyu Liu, Yunzhen Liu, Zehao Fan, Garrett Gagnon, Yayue Hou, Nan Wu, Yangwook Kang, and Liu Liu. Bandwidth-efficient adaptive mixture-of-experts via low-rank compensation, 2025d. URL https://arxiv.org/abs/2512.17073.

Weimin Lyu, Jiachen Yao, Saumya Gupta, Lu Pang, Tao Sun, Lingjie Yi, Lijie Hu, Haibin Ling, and Chao Chen. Backdooring vision-language models with out-of-distribution data, 2025. URL https://arxiv.org/abs/2410.01264.

Shweta Mahajan, Hoang Le, Hyojin Park, Farzad Farhadzadeh, Munawar Hayat, and Fatih Porikli. Attention guided alignment in efficient vision-language models, 2025. URL https://arxiv.org/abs/2511.17793.

Hancheol Park, Geonho Lee, Tairen Piao, and Tae-Ho Kim. Value-and-structure alignment for routing-consistent quantization of mixture-of-experts models, 2026. URL https://arxiv.org/abs/2606.05688.

Ruiguang Pei, Weiqing Sun, Zhihui Fu, and Jun Wang. Greedyprune: Retenting critical visual token set for large vision language models, 2025. URL https://arxiv.org/abs/2506.13166.

Haiying Sha. Sparse mixture-of-experts routing in visual diffusion transformers:diagnosis, boundary calibration and evolutionary roadmap from routing collapse to selective deadlock, 2026. URL https://arxiv.org/abs/2605.19378.

Kele Shao, Keda Tao, Kejia Zhang, Sicheng Feng, Mu Cai, Yuzhang Shang, Haoxuan You, Can Qin, Yang Sui, and Huan Wang. A survey of token compression for efficient multimodal large language models, 2026. URL https://arxiv.org/abs/2507.20198.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. Fastvid: Dynamic density pruning for fast video large language models, 2025a. URL https://arxiv.org/abs/2503.11187.

Zixu Shen, Kexin Chu, Yifan Zhang, Dawei Xiang, Runxin Wu, and Wei Zhang. Expertflow: Adaptive expert scheduling and memory coordination for efficient moe inference, 2025b. URL https://arxiv.org/abs/2510.26730.

Jiafei Song, Fengwei Zhou, Jin Qu, Wenjin Jason Li, Tong Wu, Gengjian Xue, Zhikang Zhao, Daomin Wei, Yichao Lu, and Bailin Na. Evocomp: Learning visual token compression for multimodal large language models via semantic-guided evolutionary labeling, 2026. URL https://arxiv.org/abs/2604.17087.

Rhea Sanjay Sukthanker, Arber Zela, Benedikt Staffler, Aaron Klein, Lennart Purucker, Joerg K. H. Franke, and Frank Hutter. Hw-gpt-bench: Hardware-aware architecture benchmark for language models, 2024. URL https://arxiv.org/abs/2405.10299.

Libo Sun, Po wei Harn, Peixiong He, and Xiao Qin. When does sparse moe help in vision? the role of backbone compute leverage in sparse routing, 2026. URL https://arxiv.org/abs/2605.15484.

Suraiya Tairin, Shohaib Mahmud, Haiying Shen, and Anand Iyer. emoe: Task-aware memory efficient mixture-of-experts-based (moe) model inference, 2025. URL https://arxiv.org/abs/2503.06823.

Boyuan Tian, Yihan Pang, Muhammad Huzaifa, Shenlong Wang, and Sarita Adve. Towards energy-efficiency by navigating the trilemma of energy, latency, and accuracy, 2024. URL https://arxiv.org/abs/2409.04018.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/416f9cb3276121c42eebb86352a4354a-Abstract-Conference.html.

Marcos Treviso, Ji-Ung Lee, Tianchu Ji, Betty van Aken, Qingqing Cao, Manuel R. Ciosici, Michael Hassid, Kenneth Heafield, Sara Hooker, Colin Raffel, Pedro H. Martins, André F. T. Martins, Jessica Zosa Forde, Peter Milder, Edwin Simpson, Noam Slonim, Jesse Dodge, Emma Strubell, Niranjan Balasubramanian, Leon Derczynski, Iryna Gurevych, and Roy Schwartz. Efficient methods for natural language processing: A survey, 2023. URL https://arxiv.org/abs/2209.00099.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Arya Tschand, Arun Tejusve Raghunath Rajan, Sachin Idgunji, Anirban Ghosh, Jeremy Holleman, Csaba Kiraly, Pawan Ambalkar, Ritika Borkar, Ramesh Chukka, Trevor Cockrell, Oliver Curtis, Grigori Fursin, Miro Hodak, Hiwot Kassa, Anton Lokhmotov, Dejan Miskovic, Yuechao Pan, Manu Prasad Manmathan, Liz Raymond, Tom St. John, Arjun Suresh, Rowan Taubitz, Sean Zhan, Scott Wasson, David Kanter, and Vijay Janapa Reddi. Mlperf power: Benchmarking the energy efficiency of machine learning systems from microwatts to megawatts for sustainable ai, 2025. URL https://arxiv.org/abs/2410.12032.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Haodong Wang, Qihua Zhou, Zicong Hong, and Song Guo. D2moe: Dual routing and dynamic scheduling for efficient on-device moe-based llm serving. In *Proceedings of the 31st Annual International Conference on Mobile Computing and Networking*, ACM MOBICOM '25, pp. 574–588. ACM, November 2025a. doi: 10.1145/3680207.3723493. URL http://dx.doi.org/10.1145/3680207.3723493.

Hongyu Wang, Jiayu Xu, Ruiping Wang, Yan Feng, Yitao Zhai, Peng Pei, Xunliang Cai, and Xilin Chen. Mote: Mixture of ternary experts for memory-efficient large multimodal models, 2026. URL https://arxiv.org/abs/2506.14435.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts, 2024a. URL https://arxiv.org/abs/2408.15664.

Xijun Wang, Junyun Huang, Rayyan Abdalla, Chengyuan Zhang, Ruiqi Xian, and Dinesh Manocha. Bi-vlm: Pushing ultra-low precision post-training quantization boundaries in vision-language models, 2025b. URL https://arxiv.org/abs/2509.18763.

Xu Wang, Jiangxia Cao, Zhiyi Fu, Kun Gai, and Guorui Zhou. Home: Hierarchy of multi-gate experts for multi-task learning at kuaishou, 2024b. URL https://arxiv.org/abs/2408.05430.

Ziteng Wang, Jun Zhu, and Jianfei Chen. Remoe: Fully differentiable mixture-of-experts with relu routing, 2025c. URL https://arxiv.org/abs/2412.14711.

Ziyi Wang, Haoran Wu, Yiming Rong, Deyang Jiang, Yixin Zhang, Yunlong Zhao, Shuang Xu, and Bo XU. Lvc: A lightweight compression framework for enhancing vlms in long video understanding, 2025d. URL https://arxiv.org/abs/2504.06835.

Xing Wu, Chaochen Gao, Zijia Lin, Zhongyuan Wang, Jizhong Han, and Songlin Hu. Rap: Redundancy-aware video-language pre-training for text-video retrieval, 2022. URL https://arxiv.org/abs/2210.06881.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Yufei Xue, Yushi Huang, Jiawei Shao, Lunjie Zhu, Chi Zhang, Xuelong Li, and Jun Zhang. Vlmq: Token saliency-driven post-training quantization for vision-language models, 2026. URL https://arxiv.org/abs/2508.03351.

Minghao Yan, Hongyi Wang, and Shivaram Venkataraman. Polythrottle: Energy-efficient neural network inference on edge devices, 2024. URL https://arxiv.org/abs/2310.19991.

Yuchen Yang, Yaru Zhao, Pu Yang, Shaowei Wang, and Zhi-Hua Zhou. Zipmoe: Efficient on-device moe serving via lossless compression and cache-affinity scheduling, 2026. URL https://arxiv.org/abs/2601.21198.

Zeyu Yang, Karel Adamek, and Wesley Armour. Double-exponential increases in inference energy: The cost of the race for accuracy, 2024. URL https://arxiv.org/abs/2412.09731.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Rongjie Yi, Liwei Guo, Shiyun Wei, Ao Zhou, Shangguang Wang, and Mengwei Xu. Edgemoe: Empowering sparse large language models on mobile devices. *IEEE Transactions on Mobile Computing*, 24(8):7059–7073, 2025a. doi: 10.1109/TMC.2025.3546466.

Rongjie Yi, Liwei Guo, Shiyun Wei, Ao Zhou, Shangguang Wang, and Mengwei Xu. Edgemoe: Empowering sparse large language models on mobile devices, 2025b. URL https://arxiv.org/abs/2308.14352.

Hanxun Yu, Wentong Li, Xuan Qu, Song Wang, Junbo Chen, and Jianke Zhu. Visiontrim: Unified vision token compression for training-free mllm acceleration, 2026. URL https://arxiv.org/abs/2601.22674.

Yichao Yuan, Lin Ma, and Nishil Talati. Moe-lens: Towards the hardware limit of high-throughput moe llm serving under resource constraints, 2025. URL https://arxiv.org/abs/2504.09345.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://papers.nips.cc/paper/2020/hash/c8512d142a2d849725f31a9a7a361ab9-Abstract.html.

Haichao Zhang and Yun Fu. Vqtoken: Neural discrete token representation learning for extreme token reduction in video large language models, 2025. URL https://arxiv.org/abs/2503.16980.

Hongzhi Zhang, Jingyuan Zhang, Xingguang Ji, Qi Wang, and Fuzheng Zhang. Dyntok: Dynamic compression of visual tokens for efficient and effective video understanding, 2025a. URL https://arxiv.org/abs/2506.03990.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. Sparsevlm: Visual token sparsification for efficient vision-language model inference, 2025b. URL https://arxiv.org/abs/2410.04417.

Yudi Zhang, Weilin Zhao, Xu Han, Tiejun Zhao, Wang Xu, Hailong Cao, and Conghui Zhu. Speculative decoding meets quantization: Compatibility evaluation and hierarchical framework design, 2025c. URL https://arxiv.org/abs/2505.22179.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2o: Heavy-hitter oracle for efficient generative inference of large language models, 2023. URL https://arxiv.org/abs/2306.14048.

Zhexiang Zhang, Ye Wang, Yumiao Zhao, Jiayu Xiao, Qianjing Yang, Xiangyu Wang, Jingzhe Jiang, Qizhen Weng, Ruichuan Chen, Shaohuai Shi, Adel N. Toosi, Yin Chen, and Minchen Yu. Janus: Disaggregating attention and experts for scalable moe inference, 2026. URL https://arxiv.org/abs/2512.13525.

Wenhao Zhao, Qiran Zou, Rushi Shah, and Dianbo Liu. Representation collapsing problems in vector quantization, 2024. URL https://arxiv.org/abs/2411.16550.

Zhixiong Zhao, Zukang Xu, Zhixuan Chen, and Dawei Yang. Mobie: Efficient inference of mixture of binary experts under post-training quantization, 2026. URL https://arxiv.org/abs/2604.06798.

Shuzhang Zhong, Ling Liang, Yuan Wang, Runsheng Wang, Ru Huang, and Meng Li. Adapmoe: Adaptive sensitivity-based expert gating and management for efficient moe inference. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '24, pp. 1–9. ACM, October 2024. doi: 10.1145/3676536.3676741. URL http://dx.doi.org/10.1145/3676536.3676741.

Yi Zhong, Haotong Qin, Xindong Zhang, Lei Zhang, and Guolei Sun. Breaking modality heterogeneity in low-bit quantization for large vision-language models, 2026. URL https://arxiv.org/abs/2605.19929.

Chenxi Zhou, Pengfei Cao, Jiang Li, Bohan Yu, Jinyu Ye, Jun Zhao, and Kang Liu. From signal degradation to computation collapse: Uncovering the two failure modes of llm quantization, 2026. URL https://arxiv.org/abs/2604.19884.

Yongxin Zhu, Bocheng Li, Yifei Xin, Zhihua Xia, and Linli Xu. Addressing representation collapse in vector quantized models with one linear layer, 2025. URL https://arxiv.org/abs/2411.02038.

Heqing Zou, Tianze Luo, Guiyang Xie, Victor, Zhang, Fengmao Lv, Guangcong Wang, Junyang Chen, Zhuochen Wang, Hansheng Zhang, and Huaijian Zhang. From seconds to hours: Reviewing multimodal large language models on comprehensive long video understanding, 2024. URL https://arxiv.org/abs/2409.18938.